8-2004

A Study of Hidden Markov Model

Yang Liu
University of Tennessee - Knoxville

To the Graduate Council:

I am submitting herewith a thesis written by Yang Liu entitled "A Study of Hidden Markov Model." I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Mathematics.

Jan Rosinski, Major Professor

We have read this thesis and recommend its acceptance:

Xia Chen, Balram Rajput

Accepted for the Council:
Dixie L. Thompson

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

To the Graduate Council:

I am submitting herewith a thesis written by Yang Liu entitled "A Study of Hidden Markov Model." I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Mathematics.

Jan Rosinski

_____

Major Professor

We have read this thesis

and recommend its acceptance:

Xia Chen

_____

Balram Rajput

_____

Accepted for the Council:

Anne Mayhew

_____

Vice Chancellor and

Dean of Graduate Studies

(Original signatures are on file with official student records.)

A STUDY OF HIDDEN MARKOV MODEL

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Yang Liu
August 2004

DEDICATION

This dissertation is dedicated to my parents, Jiaquan Liu and Xiaobo Yang, for consistently supporting me, inspiring me, and encouraging me.

ACKNOWLEDGMENTS

I would like to thank my parents for their consistent support and encouragement. I specially express my deepest appreciation to my advisor, Dr. Jan Rosinski, for his patient, friendly, and unfailing support over the past two years. I also want to thank the other committee members, Dr. Balram Rajput and Dr Xia Chen for their comments and suggestions.

I would also like to thank Xiaohu Tang for his help.

ABSTRACT

The purpose of this thesis is fourfold: Introduce the definition of Hidden Markov Model. Present three problems about HMM that must be solved for the model to be useful in real-world application. Cover the solution of the three basic problems; explain some algorithms in detail as well as the mathematical proof. And finally, give some examples to show how these algorithms work.

TABLE OF CONTENTS

LIST OF TABLES

1 INTRODUCTION

Many of the most powerful sequence analysis methods are now based on principles of probabilistic modeling, such as Hidden Markov Models.

Hidden Markov Models (HMMs) are very powerful and flexible. They originally emerged in the domain of speech recognition. During the past several years it has become the most successful speech model. In recent years, they are widely used as useful tools in many other fields, including molecular biology, biochemistry and genetics, as well as computer vision.

An HMM is probabilistic model composed of a number of interconnected states, each of which emits an observable output. Each state has two kinds of parameters. First, symbol emission probabilities describe the probabilities of the possible outputs from the state, and second, state transition probabilities specify the probability of moving to a new state from the current one. An observed sequence of symbols is generated by starting at some initial state and moving probabilistically from state to state until some terminal state is reached, emitting observable symbols from each state that is passed through. A sequence of states is a first order Markov chain. This state sequence is hidden, only the sequence of symbols that it emits being observable; hence the term hidden Markov model.

This paper is organized in the following manner. The definition as well as some

necessary assumptions about HMMs are explained first. The three basic problems of HMMs are discussed, and then their solutions are given. To make HMMs are useful in the real world application, we must know how to solve these problems, which are the evaluation problem, the decoding problem and the learning problem. In this part, some useful algorithm are introduced, including the forward procedure, the backward procedure, the Viterbi algorithm, the $\delta - \theta$ algorithm and the Baum-welch method. Lastly, the computer programs by which HMMs are simulated are coved.

2 WHAT IS A HIDDEN MARKOV MODEL

A Hidden Markov Model is a stochastic model comprising an unobserved Markov chain $(X_t : t = 0, 1...)$ and an observable process $(Y_t : t = 0, 1, ...)$. In this paper, we only consider the case when the observations were represented as discrete symbols chosen from a finite set, and therefore we could use a discrete probability density within each state as this model.

To define an HMM, we need some elements. The number of the hidden states is N. We denote these N states by $s_1, s_2, ..., s_N$. The number of the observable states is M. We denote them by $r_1, r_2, ..., r_M$.

Specifically, $(X_t : t = 0, 1...)$ is a Markov chain with transition probability matrix $A = [a_{ij}]$ and an initial state distribution $\pi = (\pi_i)$, where $1 \leq i, j \leq N$.

By the properties of Markov chain, we know that

$$\begin{aligned} a_{ij} &= P(X_{t+1} = s_j | X_t = s_i) \\ &= P(X_{t+1} = s_j | X_t = s_i, X_{t-1} = s_k, ..., X_0 = s_l), \quad t = 0, 1, ..., \end{aligned}$$

for every $s_i, s_j, ..., s_k$ and s_l, where $1 \leq i, j, k, l \leq N$.

The initial state distribution of the Markov chain is given by

$$\pi_i = P(X_0 = s_i).$$

For a finite observation sequence $(Y_t : t = 0, 1, ..., T)$, where T is any fixed number, we have a fundamental assumption connecting the hidden state sequence $(X_t : t = 0, 1, ..., T)$ and the observation sequence, that is statistical independence of observations $(Y_t : t = 0, 1, ..., T)$. If we formulate this assumption mathematically, we have

$$\begin{aligned} & P(Y_0 = r_{j_0}, Y_1 = r_{j_1}, ..., Y_T = r_{j_T} | X_0 = s_{i_0}, X_1 = s_{i_1}, ..., X_T = s_{i_T}) \\ = & \prod_{t=0}^{T} P(Y_t = r_{j_t} | X_t = s_{i_t}), \end{aligned} \tag{1}$$

where $1 \leq i_0, i_1, ..., i_T \leq N,\ 1 \leq j_0, j_1, ..., j_T \leq M$. To simplify the notation, we denote the event sequence $(s_{i_0}, s_{i_1}, ..., s_{i_T})$ by $\mathbf{s_0^T}$, $(r_{j_0}, r_{j_1}, ..., r_{j_T})$ by $\mathbf{r_0^T}$, and denote $(X_0, X_1, ..., X_T)$ by $\mathbf{X_0^T}$, $(Y_0, Y_1, ..., Y_T)$ by $\mathbf{Y_0^T}$. Then, we put

$$b_j(k) = P(Y_t = r_k | X_t = s_j), 1 \leq j \leq N, 1 \leq k \leq M, t = 0, 1, 2....$$

We may rewrite the formula (1) by

$$P(\mathbf{Y_0^T} = \mathbf{r_0^T} | \mathbf{X_0^T} = \mathbf{s_0^T}) = \prod_{t=0}^{T} b_{i_t}(j_t) \tag{2}$$

So far, we know a Hidden Markov Model has several components. It has a set of states $s_1, s_2, ..., s_N$, a set of output symbols $r_1, r_2, ..., r_M$, a set of transitions which have associated with them a probability and an output symbol, and a starting state. When a transition is taken, it produces an output symbol. The complicating factor is that the output symbol given is not necessarily unique to that transition, and thus it

is difficult to determine which transition was the one actually taken – and this is why they are termed "hidden". See Figure 2.1.

By the knowledge of Markov Chain, we know $P(\mathbf{X_0^T} = \mathbf{s_0^T})$, the probability of the state sequence $\mathbf{s_0^T}$,

$$\begin{aligned} & P(\mathbf{X_0^T} = \mathbf{s_0^T}) \\ = \; & P(X_0 = s_{i_0}, X_1 = s_{i_1}, ..., X_T = s_{i_T}) \\ = \; & \pi_{i_0} a_{i_0 i_1} ... a_{i_{T-1} i_T} \\ = \; & \prod_{t=0}^{T} a_{i_{t-1} i_t} \end{aligned}$$

where $a_{i_{-1} i_0} = \pi_{i_0}$. Hence, the joint probability of $\mathbf{s_0^T}$ and $\mathbf{r_0^T}$ is

$$\begin{aligned} & P(\mathbf{X_0^T} = \mathbf{s_0^T}, \mathbf{Y_0^T} = \mathbf{r_0^T}) \\ = \; & P(\mathbf{Y_0^T} = \mathbf{r_0^T} | \mathbf{X_0^T} = \mathbf{s_0^T}) \cdot P(\mathbf{X_0^T} = \mathbf{s_0^T}) \\ = \; & \prod_{t=0}^{T} [b_{i_t}(j_t) a_{i_{t-1} i_t}] \end{aligned} \tag{3}$$

The transition probability matrix A, the initial state distribution π and the matrix $B = [b_j(k)]$, $1 \leq j \leq N, 1 \leq k \leq M$, define a Hidden Markov Model completely. Therefore we can use a compact notation $\lambda = (A, B, \pi)$ to denote a Hidden Markov Model with discrete probability distribution. We may think of λ as a parameter of the Hidden Markov Model. We denote the probability given a model λ by P_λ later.

Lemma 2.1. $P_\lambda(\mathbf{Y_0^T} = \mathbf{r_0^T}) = \sum_{1 \leq i_0, i_1, ..., i_T \leq N} \prod_{t=0}^{T} a_{i_{t-1} i_t} b_{i_t}(j_t)$, where $a_{i_{-1} i_0} = \pi_{i_0}$.

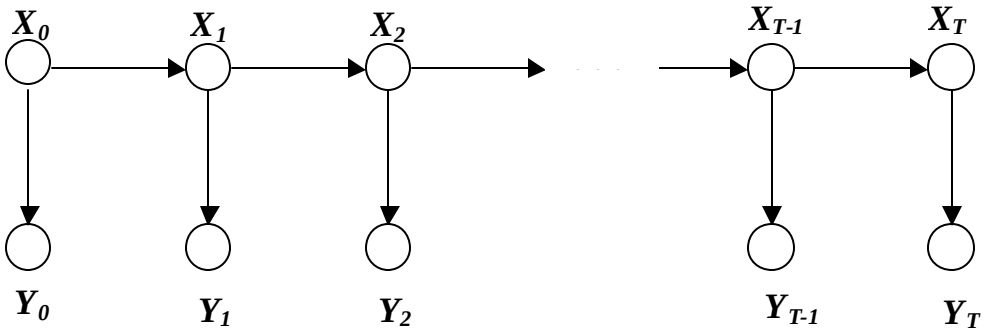

Figure 2.1 HMM

Proof: We want to compute $P_\lambda(\mathbf{Y_0^T} = \mathbf{r_0^T})$, the probability of the observation sequence given the model λ. From time 0 to time T, we consider every possible hidden state sequence, $\mathbf{s_0^T}$. Then the probability of $\mathbf{r_0^T}$ is obtained by summing the joint probability over $\mathbf{r_0^T}$ and all possible $\mathbf{s_0^T}$, that is

$$\begin{aligned} & P_\lambda(\mathbf{Y_0^T} = \mathbf{r_0^T}) \\ = & \sum_{all\ \ possible\ \mathbf{s_0^T}} P_\lambda(\mathbf{X_0^T} = \mathbf{s_0^T}, \mathbf{Y_0^T} = \mathbf{r_0^T}) \\ = & \sum_{1 \le i_0, i_1, ..., i_T \le N} \prod_{t=0}^{T} [a_{i_{t-1} i_t} b_{i_t}(j_t)]. \end{aligned} \tag{4}$$

Lemma2.1 gives us a method to compute the probability of a sequence of observations $\mathbf{r_0^T}$. But unfortunately, this calculation is computationally unfeasible. Because this formula involves on the order of $(T+1) \cdot N^{(T+1)}$ calculations. We are going to introduce some more efficient methods in the next section.

We can understand Lemma 2.1 from another point of view. A hidden Markov model consists of a set of hidden states $s_1, s_2, ..., s_N$ connected by directed edges. Each state assigns probabilities to the characters of the alphabet used in the observable sequence and to the edges leaving the state.

A path in an HMM, $s_{i_0}, s_{i_1}, ..., s_{i_T}$, is a sequence of states such that there is an edge from each state in the path to the next state in the path. And the probability of this path is the product of the probabilities of the edges traversed, that is $P(\mathbf{X_0^T} = \mathbf{s_0^T})$.

Each path through the HMM gives a probability distribution for each position in a string of the same length, based on the probabilities for the characters in the corresponding states. The probability of the observable sequence given a particular path is the product of the probabilities of the characters, that is

$$P(\mathbf{Y_0^T} = \mathbf{r_0^T} | \mathbf{X_0^T} = \mathbf{s_0^T}) = \prod_{t=0}^{T} b_{i_t}(j_t).$$

The probability of any sequence of characters is the sum, over all paths whose length is the same as the sequence, of the probability of the path times the probability of the sequence given the path, that is the result of Lemma 2.1. See Figure 2.2.

A Hidden Markov Model has a very similar property as a Markov process, that is given the values of X_t, the values of $Y_s, s \geq t$, do not depend on the values of $X_u, u < t$. The probability of any particular future observation of the model when its present hidden state is known exactly, is not altered by additional knowledge concerning its past hidden behavior. In formal terms, we have Lemma 2.2.

Lemma 2.2. $P_\lambda(Y_u = r_{j_u} | \mathbf{X_0^t} = \mathbf{s_0^t}) = P_\lambda(Y_u = r_{j_u} | X_t = s_{i_t}),\ u \geq t.$

Proof: Firstly, we prove this result is true when u=t.

$$\begin{aligned}
& P_\lambda(Y_t = r_{j_t} | \mathbf{X_0^t} = \mathbf{s_0^t}) \\
= & \sum_{0 \leq j_0, \ldots, j_{t-1} \leq M} P_\lambda(\mathbf{Y_0^t} = \mathbf{r_0^t} | \mathbf{X_0^t} = \mathbf{s_0^t}) \\
= & \sum_{0 \leq j_0, \ldots, j_{t-1} \leq M} P_\lambda(\mathbf{Y_0^{t-1}} = \mathbf{r_0^{t-1}} | \mathbf{X_0^{t-1}} = \mathbf{s_0^{t-1}}) \cdot P_\lambda(Y_t = r_{j_t} | X_t = s_{i_t})
\end{aligned}$$

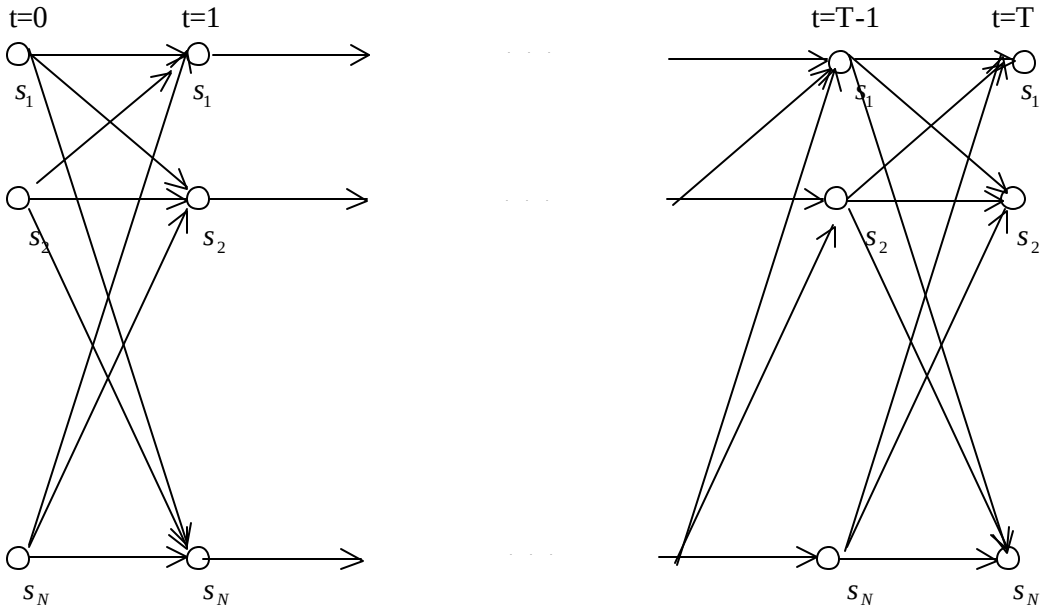

Figure 2.2 Paths of An HMM

$$= \quad 1 \cdot P_\lambda(Y_t = r_{j_t} | X_t = s_{i_t})$$

$$= \quad P_\lambda(Y_t = r_{j_t} | X_t = s_{i_t})$$

Specially, if we have $q \leq t$, we will have

$$\begin{aligned}
& P_\lambda(Y_t = r_{j_t} | \mathbf{X_q^t} = \mathbf{s_q^t}) \\
= & \sum_{0 \leq j_q, \ldots, j_{t-1} \leq M} P_\lambda(\mathbf{Y_q^t} = \mathbf{r_q^t} | \mathbf{X_q^t} = \mathbf{s_q^t}) \\
= & \sum_{0 \leq j_q, \ldots, j_{t-1} \leq M} P_\lambda(\mathbf{Y_q^{t-1}} = \mathbf{r_q^{t-1}} | \mathbf{X_q^{t-1}} = \mathbf{s_q^{t-1}}) \cdot P_\lambda(Y_t = r_{j_t} | X_t = s_{i_t}) \\
= & \quad 1 \cdot P_\lambda(Y_t = r_{j_t} | X_t = s_{i_t}) \\
= & \quad P_\lambda(Y_t = r_{j_t} | X_t = s_{i_t})
\end{aligned}$$

Then, we prove it is also true when $u > t$.

$$\begin{aligned}
& P_\lambda(Y_u = r_{j_u} | \mathbf{X_0^t} = \mathbf{s_0^t}) \\
= & \sum_{0 \leq j_0, \ldots, j_{u-1} \leq M} P_\lambda(\mathbf{Y_0^u} = \mathbf{r_0^u} | \mathbf{X_0^t} = \mathbf{s_0^t}) \\
= & \sum_{0 \leq j_0, \ldots, j_{u-1} \leq M} \sum_{0 \leq i_{t+1}, \ldots, i_u \leq N} P_\lambda(\mathbf{Y_0^u} = \mathbf{r_0^u}, \mathbf{X_{t+1}^u} = \mathbf{s_{t+1}^u} | \mathbf{X_0^t} = \mathbf{s_0^t}) \\
= & \sum_{0 \leq j_0, \ldots, j_{u-1} \leq M} \sum_{0 \leq i_{t+1}, \ldots, i_u \leq N} P_\lambda(\mathbf{Y_0^u} = \mathbf{r_0^u} | \mathbf{X_0^u} = \mathbf{s_0^u}) \cdot P_\lambda(\mathbf{X_{t+1}^u} = \mathbf{s_{t+1}^u} | X_t = s_{i_t}) \\
= & \sum_{0 \leq j_0, \ldots, j_{u-1} \leq M} \sum_{0 \leq i_{t+1}, \ldots, i_u \leq N} P_\lambda(\mathbf{Y_0^{u-1}} = \mathbf{r_0^{u-1}} | \mathbf{X_0^{u-1}} = \mathbf{s_0^{u-1}}) \cdot P_\lambda(Y_u = r_{j_u} | X_u = s_{i_u}) \\
& \cdot P_\lambda(\mathbf{X_{t+1}^u} = \mathbf{s_{t+1}^u} | X_t = s_{i_t}) \\
= & \sum_{0 \leq i_{t+1}, \ldots, i_u \leq N} P_\lambda(Y_u = r_{j_u} | X_u = s_{i_u}) \cdot P_\lambda(\mathbf{X_{t+1}^u} = \mathbf{s_{t+1}^u} | X_t = s_{i_t})
\end{aligned}$$

$$= \quad P_\lambda(Y_u = r_{j_u} | X_t = s_{i_t})$$

Lemma 2.2 will be widely used in next section to help solve the basic problems of HMM.

Remark. $(Y_t, t \geq 0)$ are not independent.

Proof: We take $T = 1$. From Lemma 2.1, we have

$$P_\lambda(\mathbf{Y_0^1} = \mathbf{r_0^1}) = \sum_{i_0, i_1 = 1}^{N} b_{i_0}(j_0) b_{i_1}(j_1) \pi_{i_0} a_{i_0 i_1}.$$

But $P_\lambda(Y_0 = r_{j_0}) = \sum_{i_0=1}^{N} b_{i_0}(j_0)\pi_{i_0}$, and

$$P_\lambda(Y_1 = r_{j_1}) = \sum_{i_1}^{N} P_\lambda(Y_1 = r_{j_1} | X_1 = s_{i_1}) P_\lambda(X_1 = s_{i_1})$$

$$= \quad \sum_{i_1}^{N} b_{i_1}(j_1) [\sum_{i_0}^{N} a_{i_0 i_1} \pi_{i_0}] = \sum_{i_0, i_1 = 1}^{N} b_{i_1}(j_1) a_{i_0 i_1} \pi_{i_0}.$$

Hence $P_\lambda(\mathbf{Y_0^1} = \mathbf{r_0^1}) \neq P_\lambda(Y_0 = r_{j_0}) \cdot P_\lambda(Y_1 = r_{j_1})$. Therefore, the sequence $(Y_t, t \geq 0)$ are not independent.

Actually, it is very natural to be understood. Because for each Y_t, it is generated by the corresponding X_t, and the hidden sequence $(X_t, t = 0, 1...)$ are not independent. There is a very easy example. We take $N = M$ and $b_i(j) = \delta_{ij}$. Then, Y_t is a Markov Chain.

3 THREE BASIC PROBLEMS OF HMMs AND THEIR SOLUTIONS

Once we have an HMM, there are three problems of interest.

3.1 The Evaluation Problem

Given a Hidden Markov Model λ and a sequence of observations $\mathbf{Y_0^T} = \mathbf{r_0^T}$, what is the probability that the observations are generated by the model, i.e. $P_\lambda(\mathbf{Y_0^T} = \mathbf{r_0^T})$. We can also view the problem as how well a given model matches a given observation sequence. By the second viewpoint, if we have several competing models, the solution to the evaluation problem will give us a best model which best matches the observation sequence.

The most straightforward way of doing this is using Lemma 2.1. But it involves a lot of calculations. The more efficient methods are called the forward procedure and the backward procedure. See[1]. We will introduce these two procedures first, then reveal the mathematical idea inside them.

3.1.1 The Forward Procedure

Fix $\mathbf{r_0^t}$ and consider the forward variable $\alpha_t(i_t)$ defined as

$$\alpha_t(i_t) = P_\lambda(\mathbf{Y_0^t} = \mathbf{r_0^t}, X_t = s_{i_t}), \qquad t = 0, 1, ..., T, \qquad 1 \le i_t \le N.$$

that is the probability of the partial observation sequence, $r_{j_0}, r_{j_1}, ..., r_{j_t}$ (until time t) and at time t the hidden state is s_{i_t}. So for the forward variable$\alpha_t(i_t)$, we only consider those paths which end at the state s_{i_t} at the time t. We can solve for $\alpha_t(i_t)$ inductively, as follows:

(1) Initialization:

$$\alpha_0(i_0) = \pi_{i_0} b_{i_0}(j_0), \qquad 1 \le i_0 \le N.$$

(2) Induction ($t = 0, 1, ..., T-1$):

$$\begin{aligned}
& \alpha_{t+1}(i_{t+1}) \\
= \;& P_\lambda(\mathbf{Y_0^{t+1}} = \mathbf{r_0^{t+1}}, X_{t+1} = s_{i_{t+1}}) \\
= \;& \sum_{1 \le i_0, ..., i_t \le N} P_\lambda(\mathbf{Y_0^{t+1}} = \mathbf{r_0^{t+1}}, \mathbf{X_0^{t+1}} = \mathbf{s_0^{t+1}}) \\
= \;& \sum_{1 \le i_0, ..., i_t \le N} P_\lambda(\mathbf{Y_0^{t+1}} = \mathbf{r_0^{t+1}} | \mathbf{X_0^{t+1}} = \mathbf{s_0^{t+1}}) \cdot P_\lambda(\mathbf{X_0^{t+1}} = \mathbf{s_0^{t+1}}) \\
= \;& \sum_{1 \le i_0, ..., i_t \le N} P_\lambda(\mathbf{Y_0^t} = \mathbf{r_0^t} | \mathbf{X_0^t} = \mathbf{s_0^t}) \cdot P_\lambda(Y_{t+1} = r_{j_{t+1}} | X_{t+1} = s_{i_{t+1}}) \\
& \cdot P_\lambda(X_{t+1} = s_{i_{t+1}} | \mathbf{X_0^t} = \mathbf{r_0^t}) \cdot P_\lambda(\mathbf{X_0^t} = \mathbf{r_0^t}) \\
= \;& [\sum_{1 \le i_0, ..., i_t \le N} P_\lambda(\mathbf{Y_0^t} = \mathbf{r_0^t}, \mathbf{X_0^t} = \mathbf{s_0^t}) \cdot P_\lambda(X_{t+1} = s_{i_{t+1}} | X_t = s_{i_t})] \cdot P_\lambda(Y_{t+1} = r_{j_{t+1}} | X_{t+1} = s_{i_{t+1}})
\end{aligned}$$

$$
\begin{aligned}
&= [\sum_{1\le i_t\le N} P_\lambda(\mathbf{Y_0^t} = \mathbf{r_0^t}, X_t = s_t)\cdot P_\lambda(X_{t+1} = s_{i_{t+1}}|X_t = s_{i_t})]\cdot P_\lambda(Y_{t+1} = r_{j_{t+1}}|X_{t+1} = s_{i_{t+1}}) \\
&= [\sum_{i_t=1}^{N} \alpha_t(i_t)a_{i_t i_{t+1}}]b_{i_{t+1}}(j_{t+1})
\end{aligned}
$$

(3) Termination:

$$
\begin{aligned}
& P_\lambda(\mathbf{Y_0^T} = \mathbf{r_0^T}) \\
= & \sum_{i_T=1}^{N} P_\lambda(\mathbf{Y_0^T} = \mathbf{r_0^T}, X_T = s_{i_T}) \\
= & \sum_{i_T=1}^{N} \alpha_T(i_T)
\end{aligned}
$$

If we examine the computation involved in the calculation of $\alpha_t(i_t)$, we see that it requires on the order of $N^2 \cdot (T+1)$ calculations, rather than $(T+1)\cdot N^{(T+1)}$ as required by the direct calculation. Hence, the forward probability calculation is more efficient than the direct calculation.

In similar, we have another method for the Evaluation Problem. It is called backward procedure.

3.1.2 The Backward Procedure

We consider a backward variable $\beta_t(i_t)$ defined as,

$$
\beta_t(i_t) = P_\lambda(\mathbf{Y_{t+1}^T} = \mathbf{r_{t+1}^T}|X_t = s_{i_t}) \quad 0 \le t \le T-1,\ 1 \le i_t \le N.
$$

That is the probability of the partial observation sequence from time $t+1$ to the end, given the hidden state is s_{i_t} at time t. Again we can solve for $\beta_t(i_t)$ inductively, as follows:

(1) Initialization:

To make this procedure work for $t = T-1$, we arbitrarily define $\beta_T(i_T)$ to be 1 in the initialization step (1).

$$\beta_T(i_T) = 1.$$

(2) Induction $(t = 0, 1, ..., T-1)$:

$$\begin{aligned}
& \beta_t(i_t) \\
= \; & P_\lambda(\mathbf{Y_{t+1}^T} = \mathbf{r_{t+1}^T} | X_t = s_{i_t}) \\
= \; & \sum_{i_{t+1}=1}^{N} P_\lambda(\mathbf{Y_{t+1}^T} = \mathbf{r_{t+1}^T}, X_{t+1} = s_{i_{t+1}} | X_t = s_{i_t}) \\
= \; & \sum_{i_{t+1}=1}^{N} P_\lambda(\mathbf{Y_{t+1}^T} = \mathbf{r_{t+1}^T} | X_{t+1} = s_{i_{t+1}}, X_t = s_{i_t}) \cdot P_\lambda(X_{t+1} = s_{i_{t+1}} | X_t = s_{i_t}) \\
= \; & \sum_{i_{t+1}=1}^{N} P_\lambda(\mathbf{Y_{t+1}^T} = \mathbf{r_{t+1}^T} | X_{t+1} = s_{i_{t+1}}) \cdot P_\lambda(X_{t+1} = s_{i_{t+1}} | X_t = s_{i_t}) \\
= \; & \sum_{i_{t+1}=1}^{N} \sum_{1 \le i_{t+2}, ..., i_T \le N} P_\lambda(\mathbf{Y_{t+1}^T} = \mathbf{r_{t+1}^T} | \mathbf{X_{t+1}^T} = \mathbf{s_{t+1}^T}) \cdot P_\lambda(\mathbf{X_{t+2}^T} = \mathbf{s_{t+2}^T} | X_{t+1} = s_{i_{t+1}}) \\
& \cdot P_\lambda(X_{t+1} = s_{i_{t+1}} | X_t = s_{i_t}) \\
= \; & \sum_{i_{t+1}=1}^{N} P_\lambda(Y_{t+1} = r_{j_{t+1}} | X_{t+1} = s_{i_{t+1}}) \cdot P_\lambda(X_{t+1} = s_{i_{t+1}} | X_t = s_{i_t})
\end{aligned}$$

$$
\begin{aligned}
&\cdot[\sum_{1\le i_{t+2},\ldots,i_T\le N} P_\lambda(\mathbf{Y_{t+2}^T}=\mathbf{r_{t+2}^T}|\mathbf{X_{t+1}^T}=\mathbf{s_{t+1}^T})P_\lambda(\mathbf{X_{t+2}^T}=\mathbf{s_{t+2}^T}|X_{t+1}=s_{i_{t+1}})]\\
=&\sum_{i_{t+1}=1}^{N} P_\lambda(Y_{t+1}=r_{j_{t+1}}|X_{t+1}=s_{i_{t+1}})\cdot P_\lambda(\mathbf{Y_{t+2}^T}=\mathbf{r_{t+2}^T}|X_{t+1}=s_{i_{t+1}})\\
&\cdot P_\lambda(X_{t+1}=s_{i_{t+1}}|X_t=s_{i_t})\\
=&\sum_{i_{t+1}=1}^{N} a_{i_t i_{t+1}} b_{i_{t+1}}(j_{t+1})\beta_{t+1}(i_{t+1}).
\end{aligned}
$$

(3) Termination:

$$
\begin{aligned}
&P_\lambda(\mathbf{Y_0^T}=\mathbf{r_0^T})\\
=&\sum_{i_0=1}^{N} P_\lambda(\mathbf{Y_0^T}=\mathbf{r_0^T}, X_0=s_{i_0})\\
=&\sum_{i_0=1}^{N} P_\lambda(\mathbf{Y_0^T}=\mathbf{r_0^T}|X_0=s_{i_0})\cdot P_\lambda(X_0=s_{i_0})\\
=&\sum_{i_0=1}^{N} P_\lambda(\mathbf{Y_1^T}=\mathbf{r_1^T}|X_0=s_{i_0})\cdot P_\lambda(Y_0=r_{j_0}|X_0=s_{i_0})\cdot P_\lambda(X_0=s_{i_0})\\
=&\sum_{i_0=1}^{N} \beta_0(i_0) b_{i_0}(j_0)\pi_{i_0}
\end{aligned}
$$

The back ward procedure requires on the order of $N^2\cdot(T+1)$ calculations, as many as the forward procedure.

3.1.3 Summary of the Evaluation Problem

We have introduced how to evaluate the probability that the observation sequence $\mathbf{Y_0^T}=\mathbf{r_0^T}$ is generated by using either the forward procedure or the backward proce-

dure. They are mode efficient than the method given in Lemma 2.1.

In fact, these two procedures are nothing but changing multiple sum, that is Lemma 2.1, to repeated sum. For example, in the forward procedure, we use the identity

$$\sum_{1\leq i_0,\ldots,i_T\leq N} * = \sum_{i_T=1}^{N} \ldots \sum_{i_0=1}^{N} *,$$

and for the backward procedure, we reverse the order of summation, that is

$$\sum_{1\leq i_0,\ldots,i_T\leq N} * = \sum_{i_0=1}^{N} \ldots \sum_{i_T=1}^{N} *.$$

From this point of view, it is obvious that other procedures are possible. For example, we can do he summation from the two ends to the middle at the same time. In some sense, this can be seen as a kind of parallel algorithm. It requires on the order of $N(N-1)\cdot(T+1)$ calculations.

3.2 The Decoding Problem

Given a model λ and a sequence of observations $\mathbf{Y_0^T} = \mathbf{r_0^T}$, what is the most likely state sequence in the model that produced the observation? That is, we want to find a hidden state sequence $\mathbf{X_0^T} = \mathbf{s_0^T}$, to maximize the probability, $P_\lambda(\mathbf{X_0^T} = \mathbf{s_0^T} | \mathbf{Y_0^T} = \mathbf{r_0^T})$, for any possible sequences $\mathbf{s_0^T}$. This is equivalent to maximize $P_\lambda(\mathbf{X_0^T} = \mathbf{s_0^T}, \mathbf{Y_0^T} = \mathbf{r_0^T})$, because the probability of $\mathbf{r_0^T}$ given a model λ, $P_\lambda(\mathbf{Y_0^T} = \mathbf{r_0^T})$ is fixed. A technique for finding this state sequence exists, based on dynamic programming methods, and

is called the Viterbi algorithm. See[1].

3.2.1 Viterbi Algorithm

Fixing s_{i_t}, we consider a variable $\delta_t(i_t)$ defined as

$$\delta_t(i_t) = \max_{1\le i_0,i_1,\ldots,i_{t-1}\le N} P_\lambda(\mathbf{X_0^{t-1}} = \mathbf{s_0^{t-1}}, X_t = s_{i_t}, \mathbf{Y_0^t} = \mathbf{r_0^t}).$$

Hence, $\delta_t(i_t)$ is the highest probability along a path, which accounts for the first t observations and ends in state s_{i_t} at time t.

By induction we have,

$$\begin{aligned}
& \delta_{t+1}(i_{t+1}) \\
= & \max_{1\le i_0,i_1,\ldots,i_t\le N} P_\lambda(\mathbf{X_0^t} = \mathbf{s_0^t}, X_{t+1} = s_{i_{t+1}}, \mathbf{Y_0^{t+1}} = \mathbf{r_0^{t+1}}) \\
= & \max_{1\le i_0,i_1,\ldots,i_t\le N} P_\lambda(\mathbf{Y_0^{t+1}} = \mathbf{r_0^{t+1}} | \mathbf{X_0^{t+1}} = \mathbf{s_0^{t+1}}) \cdot P_\lambda(\mathbf{X_0^{t+1}} = \mathbf{s_0^{t+1}}) \\
= & \max_{1\le i_0,i_1,\ldots,i_t\le N} P_\lambda(Y_{t+1} = r_{j_{t+1}} | X_{t+1} = s_{i_{t+1}}) \cdot P_\lambda(\mathbf{Y_0^t} = \mathbf{r_0^t} | \mathbf{X_0^t} = \mathbf{s_0^t}) \\
& \cdot P_\lambda(X_{t+1} = s_{i_{t+1}} | \mathbf{X_0^t} = \mathbf{s_0^t}) \cdot P_\lambda(\mathbf{X_0^t} = \mathbf{s_0^t}) \\
= & \max_{1\le i_0,i_1,\ldots,i_t\le N} P_\lambda(Y_{t+1} = r_{j_{t+1}} | X_{t+1} = s_{i_{t+1}}) \cdot P_\lambda(\mathbf{Y_0^t} = \mathbf{r_0^t}, \mathbf{X_0^t} = \mathbf{s_0^t}) \\
& \cdot P_\lambda(X_{t+1} = s_{i_{t+1}} | X_t = s_{i_t}) \\
= & [\max_{1\le i_t\le N} \max_{1\le i_0,\ldots,i_{t-1}\le N} P_\lambda(\mathbf{Y_0^t} = \mathbf{r_0^t}, \mathbf{X_0^t} = \mathbf{s_0^t}) \cdot P_\lambda(X_{t+1} = s_{i_{t+1}} | X_t = s_{i_t})] \\
& \cdot P_\lambda(Y_{t+1} = r_{j_{t+1}} | X_{t+1} = s_{i_{t+1}}) \\
= & [\max_{1\le i_t\le N} \delta_t(i_t) a_{i_t i_{t+1}}] b_{i_{t+1}}(j_{t+1}).
\end{aligned}$$

To find the hidden state sequence, we need to keep track of the argument which maximize $\delta_t(i_t)$, for every t and i_t, and they will be noted by an array $\psi_t(i_t)$.

(1) Initialization:

$$\delta_0(i_0) = \pi_{i_0} b_{i_0}(j_0), \qquad 1 \leq i_0 \leq N.$$

(2)Induction (t=1,...,T):

$$\delta_t(i_t) = \max_{1 \leq i_{t-1} \leq N} [\delta_{t-1}(i_{t-1}) a_{i_{t-1} i_t}] b_{i_t}(j_t),$$

$$\psi_t(i_t) = \arg \max_{1 \leq i_{t-1} \leq N} [\delta_{t-1}(i_{t-1}) a_{i_{t-1} i_t}].$$

(3)Termination:

$$\delta^* = \max_{1 \leq i_T \leq N} [\delta_T(i_T)],$$

$$\psi^* = \arg \max_{1 \leq i_T \leq N} [\delta_T(i_T)].$$

Finally, we will have the state sequence $i_T = \psi^*$ and $i_t = \psi_{t+1}(i_{t+1}), \quad t = T - 1, T - 2, ..., 0.$

Remark1. In the Viterbi Algorithm, $\psi_t(i_t),\ t = 0, ..., T-1$, and ψ^* are set-valued maps. This means there may be more than one best hidden state sequence. In order to distinguish them, we need more information.

Remark2. This procedure is not consistent with T increasing.

Proof: We give a counterexample. We take

$$\pi = \begin{pmatrix} 0.6 & 0.4 \end{pmatrix},$$

$$A = \begin{pmatrix} 0.5 & 0.5 \\ 0.9 & 0.1 \end{pmatrix},$$

$$B = \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{pmatrix},$$

Suppose we have an observation sequence $\mathbf{Y_0^1} = (r_1, r_2)$. Following the Viterbi Algorithm, when $T = 0$, the most likely hidden state is s_1, but when $T' = 1$, the best one is (s_2, s_1).

This means, given a fixed sequence $\mathbf{Y}$, if s_{i_0} is a solution for $T = 0$ and $(s_{i'_0}, s_{i'_1})$ is a solution for $T' = 1$, we may not have $i_0 = i'_0$. Even if s_{i_0} and $(s_{i'_0}, s_{i'_1})$ are the unique best solutions respectively.

The Viterbi Algorithm is very similar as the forward procedure that we have introduced in the above section. In the forward procedure, we define $\alpha_t(i_t)$, while here we use $\delta_t(i_t)$. The only diffence between them is we change summation to maximum. But the general ideas are same. We change multiple summation to repeated summation and multiple maximum to repeated maximum.

From this point of view, it is very natural to think how to use the idea that we have used in the backward procedure to solve the Decoding Problem. Therefore, we

introduce a new method. We do the maximum from the two ends to the middle at the same time. That is,

3.2.2 $\delta - \theta$ Algorithm

The advantage of this method is saving time.

To describe this procedure, we define another variable $\theta_t(i_t),\ t = 0, ..., T-1$.

$$\theta_t(i_t) = \max_{1 \le i_{t+1}, ..., i_T \le N} P_\lambda(\mathbf{X_{t+1}^T} = \mathbf{s_{t+1}^T}, \mathbf{Y_{t+1}^T} = \mathbf{r_{t+1}^T} | X_t = s_{i_t}).$$

$\theta_t(i_t)$ is the highest probability along a path, which accounts from time t to the end and starts at state s_{i_t} at time t.

Inductively, we have

$$\begin{aligned}
& \theta_t(i_t) \\
= & \max_{1 \le i_{t+1}, ..., i_T \le N} P_\lambda(\mathbf{X_{t+1}^T} = \mathbf{s_{t+1}^T}, \mathbf{Y_{t+1}^T} = \mathbf{r_{t+1}^T} | X_t = s_{i_t}) \\
= & \max_{1 \le i_{t+1}, ..., i_T \le N} P_\lambda(\mathbf{X_{t+2}^T} = \mathbf{s_{t+2}^T}, \mathbf{Y_{t+1}^T} = \mathbf{r_{t+1}^T} | X_{t+1} = s_{i_{t+1}}, X_t = s_{i_t}) \\
& \cdot P_\lambda(X_{t+1} = s_{i_{t+1}}, X_t = s_{i_t}) \\
= & \max_{1 \le i_{t+1}, ..., i_T \le N} P_\lambda(\mathbf{Y_{t+1}^T} = \mathbf{r_{t+1}^T} | \mathbf{X_t^T} = \mathbf{s_t^T}) \cdot P_\lambda(\mathbf{X_{t+2}^T} = \mathbf{s_{t+2}^T} | X_{t+1} = s_{i_{t+1}}) \\
& \cdot P_\lambda(X_{t+1} = s_{i_{t+1}} | X_t = s_{i_t}) \\
= & \max_{1 \le i_{t+1}, ..., i_T \le N} P_\lambda(\mathbf{Y_{t+2}^T} = \mathbf{r_{t+2}^T} | \mathbf{X_{t+2}^T} = \mathbf{s_{t+2}^T}) \cdot P_\lambda(Y_{t+1} = r_{j_{t+1}} | X_{t+1} = s_{i_{t+1}}) \\
& \cdot P_\lambda(\mathbf{X_{t+2}^T} = \mathbf{s_{t+2}^T} | X_{t+1} = s_{i_{t+1}}) \cdot P_\lambda(X_{t+1} = s_{i_{t+1}} | X_t = s_{i_t})
\end{aligned}$$

$$= \max_{1\le i_{t+1},\ldots,i_T\le N} P_\lambda(\mathbf{Y_{t+2}^T}=\mathbf{r_{t+2}^T},\mathbf{X_{t+2}^T}=\mathbf{s_{t+2}^T}|X_{t+1}=s_{i_{t+1}})\cdot P_\lambda(Y_{t+1}=r_{j_{t+1}}|X_{t+1}=s_{i_{t+1}})$$

$$\cdot P_\lambda(X_{t+1}=s_{i_{t+1}}|X_t=s_{i_t})$$

$$= \max_{1\le i_{t+1}\le N} \theta_{t+1}(i_{t+1})a_{i_ti_{t+1}}b_{i_{t+1}}(j_{t+1})$$

(1)Initialization:

$$\delta_0(i_0)=\pi_{i_0}b_{i_0}(j_0),\quad 1\le i_0\le N,$$

$$\theta_T(i_T)=1,\quad 1\le i_T\le N.$$

(2)Induction:

$$\delta_t(i_t)=\max_{1\le i_{t-1}\le N}[\delta_{t-1}(i_{t-1})a_{i_{t-1}i_t}]b_{i_t}(j_t),\quad t=1,...,T.$$

$$\psi_t(i_t)=\arg\max_{1\le i_{t-1}\le N}[\delta_{t-1}(i_{t-1})a_{i_{t-1}i_t}],\quad t=1,...,T.$$

$$\theta_t(i_t)=\max_{1\le i_{t+1}\le N}\theta_{t+1}(i_{t+1})a_{i_ti_{t+1}}b_{i_{t+1}}(j_{t+1}),\quad t=0,...,T-1.$$

$$\varphi_t(i_t)=\arg\max_{1\le i_{t+1}\le N}\theta_{t+1}(i_{t+1})a_{i_ti_{t+1}}b_{i_{t+1}}(j_{t+1}),\quad t=0,...,T-1.$$

(3)Termination:

$$\max_{1\le i_0,\ldots,i_T\le N} P_\lambda(\mathbf{X_0^T}=\mathbf{s_0^T},\mathbf{Y_0^T}=\mathbf{r_0^T})$$

$$
\begin{aligned}
&= \max_{1\le i_t\le N}\ \max_{1\le i_s\le N, s\neq t} P_\lambda(\mathbf{X_0^T}=\mathbf{s_0^T}, \mathbf{Y_0^T}=\mathbf{r_0^T}) \\
&= \max_{1\le i_t\le N}\ \max_{1\le i_s\le N, s\neq t} P_\lambda(\mathbf{Y_0^T}=\mathbf{r_0^T}|\mathbf{X_0^T}=\mathbf{s_0^T}) P_\lambda(\mathbf{X_0^T}=\mathbf{s_0^T}) \\
&= \max_{1\le i_t\le N}\ \max_{1\le i_s\le N, s\neq t} P_\lambda(\mathbf{Y_0^t}=\mathbf{r_0^t}|\mathbf{X_0^t}=\mathbf{s_0^t}) P_\lambda(\mathbf{X_0^t}=\mathbf{s_0^t}) \\
&\quad \cdot P_\lambda(\mathbf{Y_{t+1}^T}=\mathbf{r_{t+1}^T}|\mathbf{X_{t+1}^T}=\mathbf{s_{t+1}^T}) P_\lambda(\mathbf{X_{t+1}^T}=\mathbf{s_{t+1}^T}|\mathbf{X_0^t}=\mathbf{s_0^t}) \\
&= \max_{1\le i_t\le N}\Big[\max_{1\le i_s\le N, s<t} P_\lambda(\mathbf{X_0^t}=\mathbf{s_0^t}, \mathbf{Y_0^t}=\mathbf{r_0^t}) \\
&\quad \cdot \max_{1\le i_s\le N, s>t} P_\lambda(\mathbf{Y_{t+1}^T}=\mathbf{r_{t+1}^T}, \mathbf{X_{t+1}^T}=\mathbf{s_{t+1}^T}|X_{t+1}=s_{i_{t+1}})\Big] \\
&= \max_{1\le i_t\le N}[\delta_t(i_t)\cdot\theta_t(i_t)]
\end{aligned}
$$

We define $i_t^* = \arg\max_{1\le i_t\le N}[\delta_t(i_t)\cdot\theta_t(i_t)]$ and the most likely hidden sequence is

$$
i_t = \begin{cases} \psi_{t+1}(i_{t+1}) & from\ 0\ to\ t-1 \\ i_t^* & when\ time = t \\ \varphi_{t-1}(i_{t-1}) & from\ t+1\ to\ T \end{cases}
$$

3.3 The Learning Problem

Given a model λ and a sequence of observations $\mathbf{Y_0^T}=\mathbf{r_0^T}$, how should we adjust the model parameters (A, B, π) in order to maximize $P_\lambda(\mathbf{Y_0^T}=\mathbf{r_0^T})$? We face an optimization problem with restrictions. This probability $P_\lambda(\mathbf{Y_0^T}=\mathbf{r_0^T})$ is a function of the variables $\pi_i, a_{ij}, b_j(k)$, where $1\le i,j\le N, 1\le k\le N$.

Also. we may view the probability $P_\lambda(\mathbf{Y_0^T}=\mathbf{r_0^T})$ as the likelihood function of λ,

considered as a function of λ for fixed $\mathbf{r_0^T}$. Thus, for each $\mathbf{r_0^T}$, $P_\lambda(\mathbf{Y_0^T} = \mathbf{r_0^T})$ gives the probability of observing $\mathbf{r_0^T}$. We use the method of maximum likelihood, try to find that the value of λ, that is "most likely" to have produced the sequence $\mathbf{r_0^T}$.

See Lemma 2.1.

The problem we need to solve is:

$$\max_{\lambda} P_\lambda(\mathbf{Y_0^T} = \mathbf{r_0^T}) = \max_{\lambda} \sum_{1 \le i_0, i_1, \ldots, i_T \le N} \prod_{t=0}^{T} a_{i_{t-1} i_t} b_{i_t}(j_t),$$

subject to

$$\pi_i \ge 0, \sum_{i=1}^{N} \pi_i = 1,$$

$$a_{ij} \ge 0, \sum_{j=1}^{N} a_{ij} = 1, \text{for any i},$$

$$b_j(k) \ge 0, \sum_{k=1}^{M} b_j(k) = 1, \text{for any j},$$

$$\text{where } 1 \le i, j \le N,\ 1 \le k \le M \text{ and } a_{i_{-1} i_0} = \pi_{i_0}$$

To solve this problem, we need to use some results that Leonard E. Baum and George R. Sell proved in 1968. See [3].

1. Let $M \cup \partial M$ denote the manifold with boundary given by $x = (x_{ij})$ where

$$\{x_{ij} : x_{ij} \ge 0 \text{ and } \sum_{j=1}^{q_i} x_{ij} = 1\}$$

where $q_1, ..., q_k$ is a set of nonnegative integers. Let P be a homogeneous polynomial in the variable $\{x_{ij}\}$, with nonnegative coefficients. Let $\Lambda = \Lambda_P : M \to M \bigcup \partial M$ defined by $y = \Lambda_P(x)$ where

$$y_{ij} = x_{ij} \frac{\partial P}{\partial x_{ij}} [\sum_{k=1}^{q_i} x_{ik} \frac{\partial P}{\partial x_{ik}}]^{-1}.$$

Then

$$P(x) \leq P(t\Lambda_P(x) + (1-t)x), \text{where } 0 \leq t \leq 1, x \in M.$$

Here, $M = \{x_{ij} : x_{ij} > 0 \text{ and} \sum_{j=1}^{q_i} x_{ij} = 1\}, \partial M = \{x_{ij} : \exists x_{ij} = 0 \text{ and} \sum_{j=1}^{q_i} x_{ij} = 1\}$.

2. Let P be a homogeneous polynomial in the variables (x_{ij}) with positive coefficients and let $q \in M$ be an isolated local maximum of P. Then there exists a neighborhood V of q such that $\Lambda(V) \subset V$ and for every $x \in V$

$$\Lambda^n(x) \to q \text{ as } n \to \infty.$$

3.(A) The transformation Λ_P on M can be extended to be continuous on $M \cup \partial M$.

(B) Λ_P can also be continuously extended on any isolated local maximum q of P on ∂M by the definition $\Lambda_P(q) = q$.

(C) The extended transformation Λ_P still obeys the inequality

$$P(x) \leq P(t\Lambda_P(x) + (1-t)x), \text{where } 0 \leq t \leq 1.$$

Now, we come back to the Learning Problem. We define

$$P = P_\lambda(\mathbf{Y_0^T} = \mathbf{r_0^T}) = \sum_{1 \le i_0, i_1, \ldots, i_T \le N} \prod_{t=0}^{T} a_{i_{t-1} i_t} b_{i_t}(j_t).$$

Notice that P is a $2(T+1)$ order homogeneous polynomial in the variable $\pi_i, a_{ij}, b_j(k)$, with nonnegative coefficients. And these $\pi_i, a_{ij}, b_j(k)$ satisfy the conditions in the above results.

We define a new model $\overline{\lambda} = (\overline{A}, \overline{B}, \overline{\pi})$.

When $\pi_i \neq 0, a_{ij} \neq 0, b_j(k) \neq 0$,

$$\overline{\pi}_i = \frac{\pi_i \frac{\partial P}{\partial \pi_i}}{\sum_{i=1}^{N} \pi_i \frac{\partial P}{\partial \pi_i}},$$

$$\overline{a_{ij}} = \frac{a_{ij} \frac{\partial P}{\partial a_{ij}}}{\sum_{j=1}^{N} a_{ij} \frac{\partial P}{\partial a_{ij}}},$$

$$\overline{b_j(k)} = \frac{b_j(k) \frac{\partial P}{\partial b_j(k)}}{\sum_{k=1}^{M} b_j(k) \frac{\partial P}{\partial b_j(k)}}.$$

When $\pi_i = 0$, or $a_{ij} = 0$, or $b_j(k) = 0$, we define $\overline{\pi}_i = 0, \overline{a_{ij}} = 0, \overline{b_j(k)} = 0$.

Applying the above result and taking $t = 1$, we obtain

$$P(\pi_i, a_{ij}, b_j(k)) \le P(\overline{\pi_i}, \overline{a_{ij}}, \overline{b_j(k)}).$$

Also, if λ belongs to a neighborhood V of λ^*, where λ^* is a local maximum of P, we will have either 1) the initial model λ defines a critical point of P, in which

case $\overline{\lambda} = \lambda = \lambda^*$; or 2) the model $\overline{\lambda}$ is better than the model λ in the sense that $P(\overline{\lambda}) > P(\lambda)$, or we may say $\overline{\lambda}$ is more likely than λ.

The next step that we need to solve is how to represent $\overline{\pi_i}, \overline{a_{ij}}, \overline{b_j(k)}$ by using $\pi_i, a_{ij}, b_j(k)$.

We claim:

$$\begin{aligned}
\pi_i \frac{\partial P}{\partial \pi_i} &= P_\lambda(\mathbf{Y_0^T} = \mathbf{r_0^T}, X_0 = s_i), \\
a_{ij} \frac{\partial P}{\partial a_{ij}} &= \sum_{t=0}^{T-1} P_\lambda(\mathbf{Y_0^T} = \mathbf{r_0^T}, X_t = s_i, X_{t+1} = s_{i+1}), \\
b_j(k) \frac{\partial P}{\partial b_j(k)} &= \sum_{t=0, Y_t = r_k}^{T} P_\lambda(\mathbf{Y_0^T} = \mathbf{r_0^T}, X_t = s_j).
\end{aligned}$$

Proof:

From the formula

$$P = \sum_{1 \le i_0, i_1, ..., i_T \le N} \pi_{i_0} a_{i_0 i_1} ... a_{i_t i_{t+1}} ... a_{i_{T-1} i_T} b_{i_0}(j_0) ... b_{i_t}(j_t) ... b_{i_T}(j_T),$$

we may find out P is the sum of the probability of a path $s_{i_0}, s_{i_1}, ..., s_{i_T}$. Each term of the sum is a possible path. If we take partial derivative of P with respect to π_i, all the other terms will vanish except for those represent the pathes start from state s_i. So

$$\frac{\partial P}{\partial \pi_i} = \sum_{1 \le i_1, ..., i_T \le N} a_{i i_1} ... a_{i_{T-1} i_T} b_i(j_0) ... b_{i_T}(j_T),$$

and

$$\begin{aligned} & \pi_i \frac{\partial P}{\partial \pi_i} \\ = & \sum_{1 \le i_1, \ldots, i_T \le N} \pi_i a_{i i_1} \ldots a_{i_{T-1} i_T} b_i(j_0) \ldots b_{i_T}(j_T) \\ = & P_\lambda(\mathbf{Y_0^T} = \mathbf{r_0^T}, X_0 = s_i). \end{aligned}$$

In the same way, we can proof the other two results. When we take partial derivative of P with respect to a_{ij}, only those terms, which represent the pathes stay at state s_i at some time t and move to state s_j at time $t+1$, will appear. When we take partial derivative of P with respect to $b_j(k)$, all the other terms will vanish except for those represent the pathes go through the state s_j at the time t, and the observation at that is r_k.

Then we have,

$$\begin{aligned} \overline{\pi}_i &= \frac{\pi_i \frac{\partial P}{\partial \pi_i}}{\sum_{i=1}^N \pi_i \frac{\partial P}{\partial \pi_i}} = \frac{P_\lambda(\mathbf{Y_0^T} = \mathbf{r_0^T}, X_0 = s_i)}{P_\lambda(\mathbf{Y_0^T} = \mathbf{r_0^T})}, \\ \overline{a_{ij}} &= \frac{a_{ij} \frac{\partial P}{\partial a_{ij}}}{\sum_{j=1}^N a_{ij} \frac{\partial P}{\partial a_{ij}}} = \frac{\sum_{t=0}^{T-1} P_\lambda(\mathbf{Y_0^T} = \mathbf{r_0^T}, X_t = s_i, X_{t+1} = s_{i+1})}{\sum_{t=0}^{T-1} P_\lambda(\mathbf{Y_0^T} = \mathbf{r_0^T}, X_t = s_i)}, \\ \overline{b_j(k)} &= \frac{b_j(k) \frac{\partial P}{\partial b_j(k)}}{\sum_{k=1}^M b_j(k) \frac{\partial P}{\partial b_j(k)}} = \frac{\sum_{t=0, Y_t = r_k}^{T-1} P_\lambda(\mathbf{Y_0^T} = \mathbf{r_0^T}, X_t = s_j)}{\sum_{t=0}^{T-1} P_\lambda(\mathbf{Y_0^T} = \mathbf{r_0^T}, X_t = s_j)}. \end{aligned}$$

After these theoretical results, we introduce an algorithm to realize it. That is, *the Baum-Welch method:*

In order to describe the procedure, we first define $\xi_t(i_t, i_{t+1})$, the probability of

being in state s_{i_t} at time t and state $s_{i_{t+1}}$ at time t+1, given the model λ and the observation sequence,

$$\xi_t(i_t, i_{t+1}) = P_\lambda(X_t = s_{i_t}, X_{t+1} = s_{i_{t+1}} | \mathbf{Y_0^T} = \mathbf{r_0^T}).$$

From the definition of the forward and backward variables, we can write $\xi_t(i_t, i_{t+1})$ in the form,

$$\begin{aligned} & \xi_t(i_t, i_{t+1}) \\ = & \frac{\alpha_t(i_t) a_{i_t i_{t+1}} b_{i_{t+1}}(j_{t+1}) \beta_{t+1}(i_{t+1})}{P_\lambda(\mathbf{Y_0^T} = \mathbf{r_0^T})} \\ = & \frac{\alpha_t(i_t) a_{i_t i_{t+1}} b_{i_{t+1}}(j_{t+1}) \beta_{t+1}(i_{t+1})}{\sum_{i_t=1}^{N} \sum_{i_{t+1}=1}^{N} \alpha_t(i_t) a_{i_t i_{t+1}} b_{i_{t+1}}(j_{t+1}) \beta_{t+1}(i_{t+1})} \end{aligned}$$

To solve this problem, we need another variable, $\gamma_t(i_t)$, the probability of being in state s_{i_t} at time t, given the observation sequence and the model λ, $\gamma_t(i_t) = P_\lambda(X_t = s_{i_t} | \mathbf{Y_0^T} = \mathbf{r_0^T})$.

Hence we can relate $\gamma_t(i_t)$ to $\xi_t(i_t, i_{t+1})$ by summing over i_{t+1},

$$\gamma_t(i_t) = \sum_{i_{t+1}=1}^{N} \xi_t(i_t, i_{t+1}).$$

Let $i_t = i, i_{t+1} = j$, where $1 \leq i, j \leq N$. Then,

$$\gamma_t(i) = P_\lambda(X_t = s_i | \mathbf{Y_0^T} = \mathbf{r_0^T}),$$

$$\xi_t(i, j) = P_\lambda(X_t = s_i, X_{t+1} = s_j | \mathbf{Y_0^T} = \mathbf{r_0^T}).$$

If we sum $\gamma_t(i)$ over the time index t, we get a quantity which can be interpreted as the expected number of times that state s_i is visited, that is

$$\sum_{t=0}^{T-1} \gamma_t(i) = \text{expected number of transitions from } s_i,$$

similarly,

$$\sum_{t=0}^{T-1} \xi_t(i,j) = \text{expected number of transitions from } s_i \textit{ to } s_j.$$

Using these formulas, we can give a method for reestimation of the parameters of an HMM,

$$\begin{aligned}
\overline{\pi_i} &= \text{expected frequency(number of times)in state } s_i \text{ at time 0} \\
&= \gamma_0(i), \\
\overline{a_{ij}} &= \frac{\text{expected number of transitions from } s_i \text{ to } s_j}{\text{expected number of transitions from } s_i} \\
&= \frac{\sum_{t=0}^{T-1} \xi_t(i,j)}{\sum_{t=0}^{T-1} \gamma_t(i)}, \\
\overline{b_j(k)} &= \frac{\text{expected number of times in state j and observation is } s_k}{\text{expected number of times in state j}} \\
&= \frac{\sum_{t=0, Y_t=r_k}^{T-1} \gamma_t(j)}{\sum_{t=0}^{T-1} \gamma_t(j)}.
\end{aligned}$$

Therefore, we obtain a new model $\overline{\lambda} = (\overline{A}, \overline{B}, \overline{\pi})$. Based on the above procedure, if we iteratively use $\overline{\lambda}$ in place of λ and repeat the calculation, we can improve the probability of $\mathbf{Y_0^T} = \mathbf{r_0^T}$, until some limiting point is reached.

3.4 An Example

In this section, we introduce an example to show how to solve these three problems by hands or by computer.

A school teacher gave three different types of daily homework assignments: (1)took about 5 minutes to complete, (2)took about 1 hour to complete, (3)took about 3 hours to complete. And the teacher did not reveal openly his mood to his students daily, but we know that the teacher had either in a (1)good, (2)neutral, or(3) bad mood for a whole day. Meanwhile, we knew the relationship between his mood and the homework he would assign.

We use a Hidden Markov Model λ to analysis this problem. This model includes three hidden states, {1(good mood), 2(neutral mood), 3(bad mood)} and three observable states, {1(5 minutes assignment),2(1 hour assignment), 3(3 hours assignment)}. We consider the problem in a week, from Monday to Friday, using t=1, 2, 3, 4, 5. Hence, the Markov Chain in this model is $(X_t : t = 1, 2, 3, 4, 5)$, and the observable process is $(Y_t : t = 1, 2, 3, 4, 5)$. The transition probability matrix is $A = [a_{ij}]$ and the initial state distribution is $\pi = (\pi_i)$, where

$$a_{ij} = P_\lambda(X_{t+1} = i | X_t = j),$$

$$\pi_i = P_\lambda(X_1 = i),$$

$$1 \leq i, j \leq 3, \quad t = 1,\ 2,\ 3,\ 4.$$

The relationship between the teacher's mood and the assignments is given by a matrix $B = [b_j(k)]$, where

$$b_j(k) = P_\lambda(Y_t = k | X_t = j),$$

$$1 \leq j \leq 3, 1 \leq k \leq 3, \quad t = 1, 2, 3, 4, 5.$$

In this particular case, the observable sequence $\mathbf{Y_1^5} = (1, 3, 2, 1, 3)$, and the HMM $\lambda = (A, B, \pi)$, where

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} = \begin{pmatrix} 0.2 & 0.3 & 0.5 \\ 0.2 & 0.2 & 0.6 \\ 0 & 0.2 & 0.8 \end{pmatrix},$$

$$B = \begin{pmatrix} b_1(1) & b_1(2) & b_1(3) \\ b_2(1) & b_2(2) & b_2(3) \\ b_3(1) & b_3(2) & b_3(3) \end{pmatrix} = \begin{pmatrix} 0.7 & 0.2 & 0.1 \\ 0.3 & 0.4 & 0.3 \\ 0 & 0.1 & 0.9 \end{pmatrix},$$

$$\pi = \begin{pmatrix} \pi_1 & \pi_2 & \pi_3 \end{pmatrix} = \begin{pmatrix} 0.05 & 0.2 & 0.75 \end{pmatrix}.$$

To get the initial distribution π, we use some knowledge of Markov Chain, which is the stationary probability distribution.

Definition: For a Markov Chain with the transition probability matrix $\mathbf{P} = \| P_{ij} \|_{i,j=0}^{\infty}$ and the initial distribution $\pi = (\pi_i)_{i=0}^{\infty}$, where $\sum_{j=0}^{\infty} P_{ij} = 1$ for each i, and $\sum_{i=0}^{\infty} \pi_i = 1$ for each i, if we have

$$\pi_j = \sum_{i=0}^{\infty} \pi_i P_{ij}$$

for every j, then we call the set $\pi = (\pi_i)_{i=0}^{\infty}$ is a stationary probability distribution of the Markov Chain.

For the transition matrix that we used in this example, to find the stationary probability distribution, we need to solve a linear system as the following,

$$\begin{aligned} \pi_1 &= \pi_1 a_{11} + \pi_2 a_{21} + \pi_3 a_{31} \\ \pi_2 &= \pi_1 a_{12} + \pi_2 a_{22} + \pi_3 a_{32} \\ \pi_3 &= \pi_1 a_{13} + \pi_2 a_{23} + \pi_3 a_{33} \\ 1 &= \pi_1 + \pi_2 + \pi_3 \end{aligned}$$

This linear system tells us that the stationary probability distribution is $\pi = (0.05, 0.2, 0.75)$.

The Evaluation Problem:

The first question we want to ask is what is the probability that this teacher would assign this order of homework assignments. We use the forward procedure to solve

this problem.

(1)Initialization:

$$\begin{aligned}
\alpha_1(1) &= \pi_1 b_1(1) = 0.05 \cdot 0.7 = 0.035 \\
\alpha_1(2) &= \pi_2 b_2(1) = 0.2 \cdot 0.3 = 0.06 \\
\alpha_1(3) &= \pi_3 b_3(1) = 0.75 \cdot 0 = 0
\end{aligned}$$

(2)Induction:(t=1, 2, 3, 4.)

$$\begin{aligned}
\alpha_2(1) &= (\alpha_1(1) \cdot a_{11} + \alpha_1(2) \cdot a_{21} + \alpha_1(3) \cdot a_{31}) \cdot b_1(3) \\
&= (0.035 \cdot 0.2 + 0.06 \cdot 0.2) \cdot 0.1 \\
&= 0.0019 \\
\alpha_2(2) &= (\alpha_1(1) \cdot a_{12} + \alpha_1(2) \cdot a_{22} + \alpha_1(3) \cdot a_{32}) \cdot b_2(3) \\
&= 0.00675 \\
\alpha_2(3) &= (\alpha_1(1) \cdot a_{13} + \alpha_1(2) \cdot a_{23} + \alpha_1(3) \cdot a_{33}) \cdot b_3(3) \\
&= 0.04815
\end{aligned}$$

By using the same way, we can calculate the values for all forward variables $\alpha_t(i), 1 \leq t \leq 5, 1 \leq i \leq 3$. We use Table 1 to represent our result.

Table 1. The Forward Variables $\alpha_t(i)$

$\alpha_t(i)$	1	2	3
1	0.35	0.06	0
2	0.0019	0.00675	0.04815
3	0.000346	0.00462	0.004352
4	0.0006952	0.0003083	0
5	0.0000201	0.0000811	0.0006459

(3) Termination:

$$\begin{aligned} & P_\lambda(\mathbf{Y_1^5} = (1,3,2,1,3)) \\ = \;& \alpha_5(1) + \alpha_5(2) + \alpha_5(3) \\ = \;& 0.0000201 + 0.0000811 + 0.0006459 \\ = \;& 0.007471. \end{aligned}$$

The Decoding Problem:

The second question we want to ask is what did his mood curve look like most likely that week. We use the Viterbi method.

(1)Initialization:

$$\delta_1(1) \;=\; \pi_1 b_1(1) = 0.035$$

$$\delta_1(2) = \pi_2 b_2(1) = 0.06$$

$$\delta_1(3) = \pi_3 b_3(1) = 0$$

(2)Induction:

$$\begin{aligned}
\delta_2(1) &= \max_{1\le i\le 3}[\delta_1(i)a_{i1}]b_1(3) \\
&= \max[0.007, 0.012, 0]\cdot 0.1 \\
&= 0.0012 \\
\psi_2(1) &= arg\max_{1\le i\le 3}[\delta_1(i)a_{i1}] \\
&= 2 \\
\delta_2(2) &= \max_{1\le i\le 3}[\delta_1(i)a_{i2}]b_2(3) \\
&= \max[0.0105, 0.012, 0]\cdot 0.3 \\
&= 0.0036 \\
\psi_2(2) &= arg\max_{1\le i\le 3}[\delta_1(i)a_{i2}] \\
&= 2
\end{aligned}$$

Using the same method, we will have the values for all variables $\delta_t(i)$ and $\psi_t(i)$, where $1 \le t \le 5,\ 1 \le i \le 3$. See Table 2.

Table 2. The Variables $\delta_t(i)$ and $\psi_t(i)$

$\delta_t(i)$	1	2	3	$\psi_t(i)$	1	2	3
1	0.35	0.06	0	1	0	0	0
2	0.0012	0.0036	0.0324	2	2	2	2
3	0.000144	0.002592	0.002592	3	2	3	3
4	0.0003629	0.0001556	0	4	2	2	3
5	0.0000073	0.0000327	0.0001633	5	1	1	1

(3) Termination:

$$\begin{aligned} \delta^* &= \max_{1\le i\le N}[\delta_5(i)] \\ &= 0.0001633 \\ X_T &= \arg\max_{1\le i\le 3}[\delta_5(i)] \\ &= 3 \end{aligned}$$

Finally, the hidden state sequence is given by $i_t = \psi_{t+1}(i_{t+1})$, t=4, 3, 2, 1. Hence, this teacher's mood curve is (2,3,2,1,3).

The Learning Problem:

The third problem is how to adjust the model parameters (A, B, π) to maximize $P_\lambda(\mathbf{Y_1^5} = (1, 3, 2, 1, 3))$. We use the Baum-Welch method.

To use this method, we need to know the values of the backward variables $\beta_t(i_t)$.

(1)Initialization:

$$\begin{aligned}\beta_5(1) &= 1,\\ \beta_5(2) &= 1,\\ \beta_5(3) &= 1.\end{aligned}$$

(2)Induction(t=4, 3, 2, 1.):

$$\begin{aligned}\beta_4(1) &= \sum_{i_5=1}^{3} a_{1i_5} b_{i_5}(j_5)\beta_5(i_5)\\ &= a_{11}b_1(3) + a_{12}b_2(3) + a_{13}b_3(3)\\ &= 0.2 \cdot 0.1 + 0.3 \cdot 0.3 + 0.5 \cdot 0.9\\ &= 0.56,\end{aligned}$$

$$\begin{aligned}\beta_4(2) &= \sum_{i_5=1}^{3} a_{2i_5} b_{i_5}(j_5)\beta_5(i_5)\\ &= a_{21}b_1(3) + a_{22}b_2(3) + a_{23}b_3(3)\\ &= 0.2 \cdot 0.1 + 0.2 \cdot 0.3 + 0.6 \cdot 0.9\\ &= 0.62,\end{aligned}$$

Table 3. The Backward Variables $\beta_t(i)$

$\beta_t(i)$	1	2	3
1	0.00743912	0.00803384	0.00981216
2	0.0211	0.016848	0.012224
3	0.1342	0.1156	0.0372
4	0.56	0.62	0.78
5	1	1	1

$$\begin{aligned}
\beta_4(3) &= \sum_{i_5=1}^{3} a_{3i_5} b_{i_5}(j_5)\beta_5(i_5) \\
&= a_{31}b_1(3) + a_{32}b_2(3) + a_{33}b_3(3) \\
&= 0 \cdot 0.1 + 0.2 \cdot 0.3 + 0.8 \cdot 0.9 \\
&= 0.78.
\end{aligned}$$

Using the same method, we will have the values for all variables $\beta_t(i)$, where $1 \leq t \leq 5,\ 1 \leq i \leq 3$. See Table 3.

We can use this table and follow the backward procedure to get the $P_\lambda(\mathbf{Y_0^3} = (\mathbf{1}, \mathbf{3}, \mathbf{2}, \mathbf{1}, \mathbf{3}))$.

$$\begin{aligned}
& P_\lambda(\mathbf{Y_0^3} = (\mathbf{1}, \mathbf{3}, \mathbf{2}, \mathbf{1}, \mathbf{3})) \\
= & \sum_{i_1=1}^{3} \beta_1(i_1) b_{i_1}(j_1) \pi_{i_1}
\end{aligned}$$

$$= \beta_1(1)b_1(1)\pi_1 + \beta_1(2)b_2(1)\pi_2 + \beta_1(3)b_3(1)\pi_3$$

$$= 0.0007423996.$$

We can compare this result with the one we get before. Then, the next step is to get the values of $\xi_t(i,j)$ and $\gamma_t(i)$, where $1 \le t \le 4,\ 1 \le i,j \le 3$. See Table 4.

After we get these tables, it is easy to get a new model $\overline{\lambda}$.

$$\overline{A} = \begin{pmatrix} \overline{a_{11}} & \overline{a_{12}} & \overline{a_{13}} \\ \overline{a_{21}} & \overline{a_{22}} & \overline{a_{23}} \\ \overline{a_{31}} & \overline{a_{32}} & \overline{a_{33}} \end{pmatrix} = \begin{pmatrix} 0.0896 & 0.2191 & 0.6913 \\ 0.3254 & 0.2373 & 0.4373 \\ 0 & 0.8091 & 0.1909 \end{pmatrix},$$

$$\overline{B} = \begin{pmatrix} \overline{b_1(1)} & \overline{b_1(2)} & \overline{b_1(3)} \\ \overline{b_2(1)} & \overline{b_2(2)} & \overline{b_2(3)} \\ \overline{b_3(1)} & \overline{b_3(2)} & \overline{b_3(3)} \end{pmatrix} = \begin{pmatrix} 0.8825 & 0.0630 & 0.0545 \\ 0.5632 & 0.3602 & 0.0767 \\ 0 & 0.2157 & 0.7843 \end{pmatrix},$$

$$\overline{\pi} = \begin{pmatrix} \overline{\pi_1} & \overline{\pi_2} & \overline{\pi_3} \end{pmatrix} = \begin{pmatrix} 0.3507 & 0.6493 & 0 \end{pmatrix}.$$

Therefore, we get a new model $\overline{\lambda}$, which is better than λ. We may check it by using $\overline{\lambda}$ to solve the evaluation problem again. And the result is $P_\lambda(\mathbf{Y_1^3} = (\mathbf{1},\mathbf{3},\mathbf{2},\mathbf{1},\mathbf{3})) = 0.0208$.

Table 4. The Variables $\xi_t(i,j)$ and $\gamma_t(i)$

(a)$\xi_t(i,j)$

$\xi_1(i,j)$	1	2	3
1	0.0199	0.0715	0.2593
2	0.0341	0.0817	0.5335
3	0	0	0

$\xi_2(i,j)$	1	2	3
1	0.0137	0.0355	0.0048
2	0.0488	0.0841	0.0203
3	0	0.5998	0.1930

$\xi_3(i,j)$	1	2	3
1	0.0365	0.0260	0
2	0.4879	0.2315	0
3	0	0.2181	0

$\xi_4(i,j)$	1	2	3
1	0.0187	0.0843	0.4214
2	0.0082	0.0249	0.2242
3	0	0	0

(b)$\gamma_t(i)$

$\gamma_t(i)$	1	2	3
1	0.3507	0.6493	0
2	0.054	0.1532	0.7928
3	0.0625	0.7194	0.2181
4	0.5244	0.2573	0

4 HMMs Analysis by Matlab

4.1 How to generate A, B and π

Firstly, let us discuss how to get an initial distribution π. Here, we only consider that we have finite many states. Suppose we have N hidden states. Then the initial distribution $\pi = (\pi_1, \pi_2, \cdots, \pi_N)$. We can think these π's as N parameters of a multinomial distribution.

When we estimate the parameters for the binomail distribution by using the Bayesian medols, we usually use the two-parameter beta family as the prior π. This class of distribution has the remarkable property that the resulting posterior distributions are again beta distribution. We call this property *conjugate*. So it is very natural for us to think what is a conjugate prior for the multinomial. The answer is the Dirichlet distribution. In some sense, the Dirichlet distribution is an extension of beta distribution to the high dimensional space.

Definition: The Dirichlet distribution $D(\alpha)$, where $\alpha = (\alpha_1, ..., \alpha_r)$, $\alpha_j > 0$, $1 \leq j \leq r$, is supported by a *(r-1)*-dimensional simplex Δ,

$$\Delta = \{(u_1, ..., u_{r-1}) : \sum_{i=1}^{r-1} u_i \leq 1, 0 \leq u_i \leq 1, i = 1, ..., r-1\}.$$

and the probability density is given by

$$f(u_1,...,u_{r-1}) = \begin{cases} \frac{\Gamma(\sum_{j=1}^{r}\alpha_j)}{\prod_{j=1}^{r}\Gamma(\alpha_j)} \prod_{j=1}^{r-1} u_j^{\alpha_j-1}(1-\sum_{j=1}^{r-1}u_j)^{\alpha_r-1} & if \quad (u_1,...,u_{r-1}) \in \Delta \\ \\ 0 & if \quad (u_1,...,u_{r-1}) \in R^{r-1}\backslash\Delta \end{cases}$$

Lemma 4.1. Let $N = (N_1,...,N_r)$ has the multinomial distribution $M(n,\theta)$, where $\theta = (\theta_1,...,\theta_r), 0 < \theta_j < 1, \sum_{j=1}^{r}\theta_j = 1$. If the prior distribution $\pi(\theta)$ for θ is $D(\alpha)$, then the posterior distribution $\pi(\theta|N=n)$ is $D(\alpha+n)$, where $\alpha = (\alpha_1,...,\alpha_r), n = (n_1,...,n_r)$.

Proof: For the vector $\theta = (\theta_1,...,\theta_r)$, we can rewrite it by $\theta = (\theta_1,...,\theta_{r-1})$ and define $\theta_r = 1 - \sum_{j=1}^{r-1}\theta_j$. Then we can write the prior $\pi(\theta)$ out,

$$\pi(\theta) = \frac{\Gamma(\sum_{j=1}^{r}\alpha_j)}{\prod_{j=1}^{r}\Gamma(\alpha_j)} \prod_{j=1}^{r-1}\theta_j^{\alpha_j-1}(1-\sum_{j=1}^{r-1}\theta_j)^{\alpha_r-1}.$$

By the knowledge from Bayes' rule, and define $\Delta = \{(t_1,...,t_{r-1}) : \sum_{i=1}^{r-1}t_i \le 1, 0 \le t_i \le 1, i = 1,...,r-1\}$ we have

$$\begin{aligned} \pi(\theta|N=n) &= \frac{\pi(\theta)P(N=n|\theta)}{\int_\Delta \pi(t)P(N=n|t)d(t_1,...,t_{r-1})} \\ &= C\prod_{j=1}^{r-1}\theta_j^{\alpha_j+n_j-1}(1-\sum_{j=1}^{r-1}\theta_j)^{\alpha_r+n_r-1} \end{aligned}$$

The proportionality constant C, which depends on α and n only, must be $\frac{\Gamma(\sum_{j=1}^{r}(\alpha_j+n_j))}{\prod_{j=1}^{r}\Gamma(\alpha_j+n_j)}$, and the posterior distribution of θ given n is $D(\alpha+n)$.

Lemma 4.2. See [9]. Let d be an integer number greater than 1. Let $X_1, ..., X_d$ be random variables distributed according to gamma distribution with corresponding parameters(α, γ_i), and suppose that the sequence $\{X_1, ..., X_d\}$ is independent. Then the random vector

$$(Y_1, ..., Y_d) = \frac{1}{X_1 + ... + X_d}(X_1, ..., X_d)$$

has the Dirichlet distribution.

Proof: For the random vector $(Y_1, ..., Y_d)$, we have $Y_d = 1 - \sum_{j=1}^{d-1} Y_j$. Therefor, the distribution of $(Y_1, ..., Y_d)$ is actually supported by a *(d-1)*-dimensional simplex Δ_0,

$$\Delta_0 = \{(y_1, ..., y_{d-1}) : \sum_{i=1}^{d-1} y_i \leq 1, 0 \leq y_i \leq 1, i = 1, ..., d-1\}.$$

For a Borel subset A of $\Delta = \{(y_1, ..., y_d) : \sum_{i=1}^{d} y_i = 1, 0 \leq y_i \leq 1, i = 1, ..., d\}$, let

$$B = \{(x_1, ..., x_d) : \frac{1}{x_1 + ... + x_d}(x_1, ..., x_d) \in \Delta\}.$$

Because the sequence $\{X_1, ..., X_d\}$ is independent, we can write out the probability distribution for the random vector $(X_1, ..., X_d)$,

$$P(\{\omega : \frac{(X_1(\omega), ..., X_d(\omega))}{X_1(\omega) + ... + X_d(\omega)} \in \Delta\}) = \int_B \prod_{i=1}^{d} \frac{a^{\gamma_i} x_i^{\gamma_i - 1} e^{-a x_i}}{\Gamma(\gamma_i)} d(x_1, ..., x_d),$$

where $d(x_1, ..., x_d)$ indicates that the integration is with respect to Lebesgue measure in R^d.

We make a change of variables:

$$s = x_1 + ... + x_d, \quad y_i = \frac{x_i}{s}, \quad 1 \leq i \leq d-1.$$

The Jacobian of $(x_1, ..., x_d)$with respect to $(y_1, ..., y_{d-1}, s)$ is s^{d-1}. After some calculations, the above integral equals

$$\int_C \frac{\Gamma(\sum_{j=1}^d \gamma_j)}{\prod_{j=1}^d \Gamma(\gamma_j)} \prod_{j=1}^{d-1} y_j^{\gamma_j - 1}(1 - \sum_{j=1}^{d-1} y_j)^{\gamma_d - 1} d(y_1, ..., y_{d-1}),$$

$$C = \{(y_1, ..., y_{d-1}) : (y_1, ..., y_{d-1}, 1 - y_1 - ... - y_{d-1}) \in \Delta)\}.$$

Thus we have a density for $\frac{1}{X_1 + ... + X_d}(X_1, ..., X_d)$ with respect to Lebesgue measure on Δ_0.

Lemma 4.3. Let X be a real-valued random variable with continuous distribution function F. Then, $F(X)$ is uniform distributed on $[0, 1]$. On the other hand, let Y be a real-valued random variable with uniformed distribution on $[0, 1]$, then $F^{-1}(Y)$ has the distribution F.

Proof: Let us prove the second part of this lemma. Define $X = F^{-1}(Y)$.

$$P(X \leq x) = P(F^{-1}(Y) \leq x) = P(Y \leq F(x)),$$

Since Y is uniform distributed on [0,1], we have the above probability is $F(x)$ which means X has the distribution F.

We use Lemma 4.2 and take $\alpha = 1, \gamma_i = 1, 1 \leq i \leq d$. Therefore $X_1, ..., X_d$ have the exponential distribution with parameters 1. To get the exponential distribution randomly, we apply Lemma 4.3. Generate the random variable Y first, which has the uniform distribution on [0,1]. Then apply the function $F^{-1}(x) = -ln(1 - x)$ to Y,

where F is the distribution function of the exponential distribution. So in this way, we may get the distribution for π, and for each row of A and B.

4.2 How to test the Baum-Welch method

Idea:

Firstly, we generate a model λ, which is looked as the true value. Then use λ to generate some sample sequences $\mathbf{Y_1}, ..., \mathbf{Y_n}$. To use the Baum-Welch method, we need a seed λ_0. We can think it as an initial value. For this seed λ_0 and each sequence $\mathbf{Y_i}$, we run the program to get the new model $\lambda new_i = (Anew_i, Bnew_i, \pi new_i)$, where $1 \leq i \leq n$. Finally, we get an estimate of λ, which is $\lambda new = (Anew, Bnew, \pi new)$,

$$Anew = \frac{\sum_{i=1}^{n} Anew_i}{n},$$
$$Bnew = \frac{\sum_{i=1}^{n} Bnew_i}{n},$$
$$\pi new = \frac{\sum_{i=1}^{n} \pi new_i}{n}.$$

And then we can compare this estimate λnew with the true value λ to see how the Baum-Welch method works. We also define a distant between two matrixes $A = (a_{ij})$ and $B = (b_{ij})$ by $d(A, B)$, where

$$d(A,B)=\sum_{i=1}^{N}\sum_{j=1}^{M}(a_{ij}-b_{ij})^2,$$

$$1\le i\le N, 1\le j\le M.$$

For example, we take $N=M=3$, $T=20$, and $n=10,20,30$ respectively.

Step1: λ ***and*** λ_0***.***

In this step we generate a true value λ and a seed λ_0.

$$\lambda=\left(\begin{array}{ccc}\pi, & A, & B\end{array}\right),$$

$$\pi=\left(\begin{array}{ccc}0.1540 & 0.5136 & 0.3323\end{array}\right),$$

$$A=\left(\begin{array}{ccc}0.1627 & 0.1570 & 0.6803\\ 0.0081 & 0.4129 & 0.5792\\ 0.4941 & 0.4503 & 0.0556\end{array}\right),$$

$$B=\left(\begin{array}{ccc}0.3507 & 0.3559 & 0.2933\\ 0.4116 & 0.3365 & 0.2519\\ 0.9728 & 0.0233 & 0.0039\end{array}\right),$$

$$\lambda_0 = \begin{pmatrix} \pi_0, & A_0, & B_0 \end{pmatrix},$$

$$\pi_0 = \begin{pmatrix} 0.2502 & 0.3791 & 0.3707 \end{pmatrix},$$

$$A_0 = \begin{pmatrix} 0.2552 & 0.0042 & 0.7405 \\ 0.1370 & 0.5924 & 0.2706 \\ 0.1638 & 0.4353 & 0.4009 \end{pmatrix},$$

$$B_0 = \begin{pmatrix} 0.4532 & 0.2700 & 0.2768 \\ 0.2448 & 0.5984 & 0.1568 \\ 0.1116 & 0.0098 & 0.8786 \end{pmatrix}.$$

Step2: get Y(i), take $n = 30, 1 \leq i \leq n$***.*** See Figure 4.1

Step3: Get the new model λnew***.***

Step3 is divided into two parts.

In the first part, we use the seed λ_0 to run the program. In the second part, we take the seed equals to λ.

Y1=	1	1	1	1	3	2	2	1	3	2	3	2	2	3	2	2	2	3	1	3
Y2=	3	2	2	2	2	2	2	2	2	2	2	3	3	2	2	3	3	3	3	3
Y3=	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
Y4=	1	1	3	2	2	2	2	2	1	3	2	2	2	2	2	2	2	3	2	2
Y5=	3	2	2	3	3	3	3	3	2	3	3	3	3	3	3	2	2	2	2	2
Y6=	1	1	1	1	1	1	1	1	2	2	1	1	1	1	1	2	2	1	2	1
Y7 =	1	1	2	2	2	3	2	3	3	3	3	2	3	1	1	1	1	1	1	1
Y8 =	3	3	2	2	3	2	3	3	2	2	1	1	3	2	3	2	2	3	2	2
Y9 =	2	2	2	3	3	3	2	2	3	2	2	3	3	3	2	2	3	2	3	3
Y10 =2		2	3	3	2	3	2	2	2	2	3	2	3	2	3	3	3	2	2	2
Y11 =2		2	3	3	1	2	3	3	1	2	1	2	1	3	1	2	1	3	3	1
Y12 = 1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Y13 = 1		1	3	1	1	1	1	1	1	1	1	1	1	1	1	3	1	3	1	1
Y14 = 1		3	2	1	3	1	3	1	2	1	2	2	1	3	1	2	1	2	2	2
Y15 = 2		1	2	1	2	1	2	1	2	1	1	1	2	1	2	2	1	2	1	2
Y16 = 2		1	2	2	1	2	2	2	1	2	2	1	1	1	1	2	2	2	1	1
Y17 = 2		2	2	1	2	1	2	1	2	2	1	2	1	2	1	2	1	2	1	2
Y18 = 2		2	1	2	1	2	2	2	2	2	1	2	1	2	2	1	2	1	2	1
Y19 = 1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Y20 = 1		1	2	1	2	1	2	1	1	2	1	1	1	1	1	1	1	1	1	1
Y21 = 1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Y22 = 2		2	3	3	1	2	3	3	1	2	1	2	1	3	1	2	1	3	3	1
Y23 =1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Y24 =1		1	3	1	1	1	1	1	1	1	1	1	1	1	1	3	1	3	1	1
Y25 =1		3	2	1	3	1	3	1	2	1	2	2	1	3	1	2	1	2	2	2
Y26 =2		1	2	1	2	1	2	1	2	1	1	1	2	1	2	2	1	2	1	2
Y27 =2		1	2	2	1	2	2	2	1	2	2	1	1	1	1	2	2	2	1	1
Y28 =2		2	2	1	2	1	2	1	2	2	1	2	1	2	1	2	1	2	1	2
Y29 =2		2	1	2	1	2	2	2	2	2	1	2	1	2	2	1	2	1	2	1
Y30 =1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figure 4.1 The generated observable sequences Y_i, $1 \le i \le 30$.

The reason that we want to do this is to show that the Buam-Welch method only works when the initial model λ_0 is in a neighborhood V of λ_q, where λ_q is a local maximum of the function $P = P_\lambda(\mathbf{Y_0^T} = \mathbf{r_0^T})$. If we use the seed λ_0, we will see that we can not get a good estimate of λ. The reason is λ_0 is not in a small enough neighborhood of λ. While in the second part, if we use λ as the seed, we will get good result. We think λ must be a local maximum, and it is in any neighborhood of itself. So this seed satisfies the condition in the Baum-Welch method.

PartI: use $\lambda_0 = (A_0, B_0, \pi_0)$ as the seed.

We use sample $\{Y_1, ..., Y_{10}\}, \{Y_1, ..., Y_{20}\}, \{Y_1, ..., Y_{30}\}$ respectively to get three new models $\lambda new_{1-10}, \lambda new_{1-20}, \lambda new_{1-30}$, see Figure 4.2. And then calculate the distances between them and λ. From Table 5, we can see the distances between the new models and λ are big even if we increase the sample size. This result means we can not get a good estimate from the seed λ_0.

Table 5. The Distances, Seed λ_0

	d$(\pi, \pi new)$	d$(A, Anew)$	d$(B, Bnew)$
λnew_{1-10}	0.0714	0.4708	1.9702
λnew_{1-20}	0.0575	0.3537	0.7997
λnew_{1-30}	0.0737	0.3305	0.5618

$lnew_{1-10} = (Anew_{1-10}, Bnew_{1-10}, pnew_{1-10})$

$pnew_{1-10}$ = 0.3228 0.3095 0.3677

$Anew_{1-10}$ = 0.3725 0.0006 0.6269
0.1888 0.5870 0.2242
0.1334 0.5427 0.3239

$Bnew_{1-10}$ = 0.4944 0.1637 0.3420
0.0657 0.8052 0.1291
0.1015 0.0023 0.8962

$lnew_{1-20} = (Anew_{1-20}, Bnew_{1-20}, pnew_{1-20})$

$pnew_{1-20}$ =0.3497 0.4090 0.2413

$Anew_{1-20}$ = 0.3191 0.0026 0.6783
0.2324 0.5686 0.1990
0.2630 0.5097 0.2273

$Bnew_{1-20}$ = 0.5254 0.3016 0.1730
0.2926 0.6425 0.0648
0.4268 0.0118 0.5614

$lnew_{1-30} = (Anew_{1-30}, Bnew_{1-30}, pnew_{1-30})$

$pnew_{1-30}$ = 0.3692 0.4522 0.1787

$Anew_{1-30}$ = 0.2998 0.0032 0.6970
0.2500 0.5593 0.1907
0.3074 0.4948 0.1978

$Bnew_{1-30}$ = 0.5432 0.3401 0.1167
0.3780 0.5786 0.0434
0.5356 0.0147 0.4497

Figure 4.2 Three new models corresponding sample $\{Y_1,....,Y_{10}\}, \{Y_1,...,Y_{20}\}, \{Y_1,...,Y_{30}\}$, as the seed is l_0.

Table 6. The Distances, Seed λ

	$\mathrm{d}(\pi, \pi new)$	$\mathrm{d}(A, Anew)$	$\mathrm{d}(B, Bnew)$
λnew_{1-10}	0.0415	0.7991	0.6868
λnew_{1-20}	0.0024	0.1736	0.1838
λnew_{1-30}	0.0113	0.0760	0.1058

PartII: use $\lambda = (A, B, \pi)$ as the seed.

In the same way, we use sample $\{Y_1, ..., Y_{10}\}, \{Y_1, ..., Y_{20}\}, \{Y_1, ..., Y_{30}\}$ respectively to get three new models $\lambda new_{1-10}, \lambda new_{1-20}$,and λnew_{1-30}. The results are listed in Figure 4.3. And the distances between them and λ are listed in the Table 6.

This table indicates that we get a good estimate of λ, which is very close to λ.

Discussion

When we think the $P_\lambda(\mathbf{Y_0^T} = \mathbf{r_0^T})$ as a function of λ, it may have more than one local maximum value, for example P_1 and P_2 corresponding λ_1 and λ_2. By the Baum-Welch method, we may get a good estimate of one of them, only if the initial model we used is in a neighborhood of one of them, and the closer the better. The true value λ we used here may not be a local maximum point of the function $P_\lambda(\mathbf{Y_0^T} = \mathbf{r_0^T})$. To get each new model $\lambda new_i, 1 \leq i \leq 30$, we run the program 10 times. Maybe it is not enough, because we do not know the convergence speed.

$lnew_{1-10} = (Anew_{1-10}, Bnew_{1-10}, pnew_{1-10})$

$pnew_{1-10} =$ 0.1875 0.3558 0.4567

$Anew_{1-10} =$ 0.2911 0.4510 0.2579
0.0095 0.8775 0.1130
0.6813 0.2300 0.0887

$Bnew_{1-10} =$ 0.3320 0.3531 0.3149
0.0523 0.4414 0.5063
0.4381 0.4525 0.1094

$lnew_{1-20} = (Anew_{1-20}, Bnew_{1-20}, pnew_{1-20})$

$pnew_{1-20} =$ 0.1239 0.5517 0.3244

$Anew_{1-20} =$ 0.1806 0.3170 0.5023
0.0088 0.6004 0.3909
0.6275 0.2868 0.0857

$Bnew_{1-20} =$ 0.3021 0.4588 0.2391
0.2292 0.4794 0.2914
0.7138 0.2304 0.0558

$lnew_{1-30} = (Anew_{1-30}, Bnew_{1-30}, pnew_{1-30})$

$pnew_{1-30} =$ 0.1065 0.5978 0.2957

$Anew_{1-30} =$ 0.1470 0.2701 0.5829
0.0087 0.5132 0.4781
0.6074 0.3099 0.0828

$Bnew_{1-30} =$ 0.3184 0.4677 0.2139
0.2905 0.4898 0.2197
0.8057 0.1564 0.0379

Figure 4.3 Three new models corresponding sample $\{Y_1,....,Y_{10}\}, \{Y_1,...,Y_{20}\}, \{Y_1,...,Y_{30}\}$, as the seed is l .

REFERENCES

References

[1] L. Rabinber : A tutorial on Hidden Markov Model and selected applicatopns in speech recognition , Proc. IEEE 77(2), 257-286 (1989)

[2] L. E. Baum and T. Prtrie : Statistical inference for probabilistic functions of finite state Markov Chains, Ann. Math. Stat., vol. 37, 1554-1563 (1966)

[3] L. E. Baum, G. R. Sell : Growth functions for transformations on manifolds, Pac. J. Math., vol. 27, no.2, 211-227 (1968)

[4] L. E. Baum, T. Prtrie, G. Soules, and N. Weiss: A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chain, Ann. Math. Stat., vol. 41. 164-171 (1970)

[5] J. K. Baker : The dragon system - An overview, IEEE Trans. Acoust. Speech Signal Processing, vol. ASSP-23, no. 1, 24-29, Feb. (1975)

[6] A. P. Dempster, N. M. Laird, and D. B. Rubin :Maximum likelihood fromincomplete data via the EM algorithm, J. Roy. Stat. Soc., vol. 39, no. 1, 1-38 (1977)

[7] Howard M. Taylor, Samuel Karlin: An introduction to stochastic Modeling. Revised Edition

[8] J.Hoffmann-Jorgensen: Probability with a view toward statistics. VolumeI, Volume2

[9] Bert Fristedt, Lawrence Gray: A modern approach to probability theory

APPENDICES

APPENDIX A:
The New Models Generated in Section 4.2, Step 3.

Part I, Seed l_0.

Pinew1 =

1.0000	0.0000	0.0000

Anew1 =

0.5045	0.0000	0.4955
0.1208	0.5000	0.3792
0.1946	0.8054	0.0000

Bnew1 =

1.0000	0.0000	0.0000
0.0067	0.9933	0.0000
0.0000	0.0000	1.0000

Pinew2 =

0.0000	0.0000	1.0000

Anew2 =

0.0783	0.0000	0.9217
0.1628	0.8332	0.0040
0.1784	0.4667	0.3549

Bnew2 =

0	0.0040	0.9960
0	1.0000	0.0000
0	0.0000	1.0000

Pinew3 =

0.2296	0.0956	0.6749

Anew3 =

0.1452	0.0007	0.8541
0.1602	0.1978	0.6420
0.1487	0.1129	0.7384

Bnew3 =

0	0	1
0	0	1
0	0	1

Pinew4 =

1.0000	0.0000	0.0000

Anew4 =

0.3333	0.0000	0.6667
0.0767	0.8499	0.0734
0.0000	1.0000	0.0000

Bnew4 =

1.0000	0.0000	0.0000
0.0000	0.9967	0.0033
0.0000	0.0000	1.0000

Pinew5 =

0.0000	0.0000	1.0000

Anew5 =

0.5086	0.0000	0.4914
0.2728	0.7161	0.0111
0.0895	0.4330	0.4774

Bnew5 =

0	0.0065	0.9935
0	0.9962	0.0038
0	0.0003	0.9997

Pinew6 =

0.9983	0.0000	0.0017

Anew6 =

0.5871	0.0021	0.4108
0.1124	0.7974	0.0902
0.1804	0.5211	0.2985

Bnew6 =

1.0000	0.0000	0
0.4042	0.5958	0
0.9994	0.0006	0

Pinew7 =

0.0000	0.9999	0.0001

Anew7 =

0.9987	0.0000	0.0013
0.0014	0.6626	0.3360
0.1974	0.3357	0.4669

Bnew7 =

0.9963	0.0017	0.0020
0.2462	0.6148	0.1390
0.0154	0.0167	0.9679

Pinew8 =

0.0000	0.0000	1.0000

Anew8 =

0.5075	0.0000	0.4925
0.1129	0.4780	0.4091
0.0000	0.7791	0.2209

Bnew8 =

0.9474	0.0525	0.0000
0.0000	0.9500	0.0500
0.0000	0.0000	1.0000

Pinew9 =

0.0000 1.0000 0.0000

Anew9 =

0.0513 0.0000 0.9487
0.7674 0.1726 0.0600
0.0096 0.4797 0.5107

Bnew9 =

0 0.8300 0.1700
0 0.9992 0.0008
0 0.0008 0.9992

Pinew10 =

0.0000 1.0000 0.0000

Anew10 =

0.0101 0.0031 0.9868
0.1010 0.6619 0.2371
0.3354 0.4938 0.1708

Bnew10 =

0 0.7419 0.2581
0 0.9059 0.0941
0 0.0049 0.9951

Pinew11 =

0.0000 1.0000 0.0000

Anew11 =

0.0351 0.0003 0.9646
0.7956 0.1998 0.0046
0.0002 0.5222 0.4776

Bnew11 =

0.2834 0.6770 0.0396
0.5225 0.4764 0.0011
0.1918 0.0000 0.8082

Pinew12 =

0.3929 0.4295 0.1776

Anew12 =

0.5320 0.0062 0.4618
0.2160 0.6565 0.1276
0.2778 0.5189 0.2033

Bnew12 =

1 0 0
1 0 0
1 0 0

Pinew13 =

0.7816	0.2165	0.0019

Anew13 =

0.4423	0.0034	0.5542
0.1614	0.6600	0.1785
0.4428	0.5228	0.0344

Bnew13 =

0.9990	0	0.0010
0.9953	0	0.0047
0.4119	0	0.5881

Pinew14 =

1.0000	0.0000	0.0000

Anew14 =

0.0000	0.0011	0.9989
0.2239	0.7758	0.0002
0.3071	0.6929	0.0000

Bnew14 =

1.0000	0.0000	0.0000
0.3466	0.6534	0.0000
0.0000	0.1310	0.8690

Pinew15 =

0.9997	0.0003	0.0000

Anew15 =

0.0742	0.0003	0.9256
0.1492	0.1219	0.7290
0.7946	0.2054	0.0000

Bnew15 =

0.0431	0.9569	0
0.3703	0.6297	0
1.0000	0.0000	0

Pinew16 =

0.1213	0.8787	0.0000

Anew16 =

0.3199	0.0169	0.6631
0.1258	0.7331	0.1411
0.0286	0.7059	0.2655

Bnew16 =

0.4632	0.5368	0
0.2523	0.7477	0
0.9331	0.0669	0

Pinew17 =

0.0000 1.0000 0.0000

Anew17 =

0.0966 0.0000 0.9034
0.4873 0.4163 0.0964
0.9401 0.0599 0.0000

Bnew17 =

0.0000 1.0000 0
0.0001 0.9999 0
1.0000 0.0000 0

Pinew18 =

0.0000 1.0000 0.0000

Anew18 =

0.0478 0.0039 0.9482
0.2589 0.5311 0.2100
0.6166 0.3834 0.0000

Bnew18 =

0.0000 1.0000 0
0.0007 0.9993 0
0.9930 0.0070 0

Pinew19 =

0.3929 0.4295 0.1776

Anew19 =

0.5320 0.0062 0.4618
0.2160 0.6565 0.1276
0.2778 0.5189 0.2033

Bnew19 =

1 0 0
1 0 0
1 0 0

Pinew20 =

0.0770 0.1311 0.7919

Anew20 =

0.5772 0.0080 0.4148
0.1260 0.7512 0.1228
0.2395 0.6372 0.1233

Bnew20 =

0.7760 0.2240 0
0.7075 0.2925 0
0.9921 0.0079 0

Pinew21 =

0.3929 0.4295 0.1776

Anew21 =

0.5320 0.0062 0.4618
0.2160 0.6565 0.1276
0.2778 0.5189 0.2033

Bnew21 =

1 0 0
1 0 0
1 0 0

Pinew22 =

0.0000 1.0000 0.0000

Anew22 =

0.0351 0.0003 0.9646
0.7956 0.1998 0.0046
0.0002 0.5222 0.4776

Bnew22 =

0.2834 0.6770 0.0396
0.5225 0.4764 0.0011
0.1918 0.0000 0.8082

Pinew23 =

0.3929 0.4295 0.1776

Anew23 =

0.5320 0.0062 0.4618
0.2160 0.6565 0.1276
0.2778 0.5189 0.2033

Bnew23 =

1 0 0
1 0 0
1 0 0

Pinew24 =

0.7816 0.2165 0.0019

Anew24 =

0.4423 0.0034 0.5542
0.1614 0.6600 0.1785
0.4428 0.5228 0.0344

Bnew24 =

0.9990 0 0.0010
0.9953 0 0.0047
0.4119 0 0.5881

Pinew25 =

1.0000	0.0000	0.0000

Anew25 =

0.0000	0.0011	0.9989
0.2239	0.7758	0.0002
0.3071	0.6929	0.0000

Bnew25 =

1.0000	0.0000	0.0000
0.3466	0.6534	0.0000
0.0000	0.1310	0.8690

Pinew26 =

0.9997	0.0003	0.0000

Anew26 =

0.0742	0.0003	0.9256
0.1492	0.1219	0.7290
0.7946	0.2054	0.0000

Bnew26 =

0.0431	0.9569	0
0.3703	0.6297	0
1.0000	0.0000	0

Pinew27 =

0.1213	0.8787	0.0000

Anew27 =

0.3199	0.0169	0.6631
0.1258	0.7331	0.1411
0.0286	0.7059	0.2655

Bnew27 =

0.4632	0.5368	0
0.2523	0.7477	0
0.9331	0.0669	0

Pinew28 =

0.0000	1.0000	0.0000

Anew28 =

0.0966	0.0000	0.9034
0.4873	0.4163	0.0964
0.9401	0.0599	0.0000

Bnew28 =

0.0000	1.0000	0
0.0001	0.9999	0
1.0000	0.0000	0

Pinew29 =

0.0000 1.0000 0.0000

Anew29 =

0.0478 0.0039 0.9482
0.2589 0.5311 0.2100
0.6166 0.3834 0.0000

Bnew29 =

0.0000 1.0000 0
0.0007 0.9993 0
0.9930 0.0070 0

Pinew30 =

0.3929 0.4295 0.1776

Anew30 =

0.5320 0.0062 0.4618
0.2160 0.6565 0.1276
0.2778 0.5189 0.2033

Bnew30 =

1 0 0
1 0 0
1 0 0

Part II, Seed *l* .

Pinew1 =

0.0093 0.0000 0.9907

Anew1 =

0.0860 0.4879 0.4260
0.0011 0.9830 0.0159
0.9441 0.0420 0.0139

Bnew1 =

0.9244 0.0000 0.0756
0.1269 0.5440 0.3291
0.9991 0.0009 0.0000

Pinew2 =

0.0000 1.0000 0.0000

Anew2 =

0.5339 0.1646 0.3015
0.0218 0.8390 0.1392
0.8600 0.1207 0.0193

Bnew2 =

0 0.9999 0.0001
0 0.2357 0.7643
0 0.9998 0.0002

Pinew3 =

0.1972	0.7873	0.0155

Anew3 =

0.4407	0.4946	0.0647
0.0160	0.9441	0.0399
0.4861	0.5120	0.0019

Bnew3 =

0	0	1
0	0	1
0	0	1

Pinew4 =

0.0000	0.0000	1.0000

Anew4 =

0.0000	1.0000	0.0000
0.0019	0.9264	0.0717
0.6667	0.0000	0.3333

Bnew4 =

0.0000	0.0000	1.0000
0.0000	0.9304	0.0696
0.9991	0.0009	0.0000

Pinew5 =

0.0001	0.9999	0.0000

Anew5 =

0.5267	0.1897	0.2836
0.0194	0.8683	0.1123
0.8517	0.1303	0.0180

Bnew5 =

0	0.9975	0.0025
0	0.1362	0.8638
0	0.9750	0.0250

Pinew6 =

0.1430	0.0025	0.8544

Anew6 =

0.2210	0.1730	0.6060
0.0063	0.5873	0.4064
0.5448	0.3675	0.0878

Bnew6 =

0.9630	0.0370	0
0.3963	0.6037	0
0.9382	0.0618	0

Pinew7 =

0.8280 0.0000 0.1720

Anew7 =

0.2460 0.0661 0.6879
0.0064 0.9091 0.0845
0.6922 0.1867 0.1211

Bnew7 =

1.0000 0.0000 0.0000
0.0000 0.4545 0.5454
0.9999 0.0000 0.0001

Pinew8 =

0.4761 0.5239 0.0000

Anew8 =

0.2122 0.7764 0.0114
0.0002 0.8829 0.1170
0.6712 0.0531 0.2757

Bnew8 =

0.4322 0.0011 0.5667
0.0000 0.5411 0.4589
0.4449 0.5549 0.0002

Pinew9 =

0.1672 0.0077 0.8252

Anew9 =

0.2580 0.6927 0.0493
0.0073 0.9671 0.0256
0.6228 0.3680 0.0092

Bnew9 =

0 0.9503 0.0497
0 0.4494 0.5506
0 0.9887 0.0113

Pinew10 =

0.0541 0.2366 0.7093

Anew10 =

0.3865 0.4646 0.1489
0.0145 0.8682 0.1173
0.4737 0.5194 0.0068

Bnew10 =

0 0.5452 0.4548
0 0.5189 0.4811
0 0.9425 0.0575

Pinew11 =

0.1648	0.8352	0.0000

Anew11 =

0.0000	0.2190	0.7810
0.0000	0.5603	0.4397
0.6768	0.3232	0.0000

Bnew11 =

0.0000	0.8448	0.1552
0.0000	0.2753	0.7247
0.9787	0.0000	0.0213

Pinew12 =

0.1129	0.4265	0.4606

Anew12 =

0.0990	0.1083	0.7927
0.0051	0.2954	0.6995
0.4448	0.4594	0.0958

Bnew12 =

1	0	0
1	0	0
1	0	0

Pinew13 =

0.0021	0.9979	0.0000

Anew13 =

0.0015	0.0739	0.9246
0.0003	0.1196	0.8801
0.4720	0.4494	0.0786

Bnew13 =

0.3296	0	0.6704
0.9620	0	0.0380
1.0000	0	0.0000

Pinew14 =

0.0000	0.0000	1.0000

Anew14 =

0.0001	0.2709	0.7290
0.0000	0.4392	0.5608
0.6175	0.3825	0.0000

Bnew14 =

0.0000	0.1925	0.8075
0.0000	0.9983	0.0017
1.0000	0.0000	0.0000

Pinew15 =

0.2079	0.7921	0.0000

Anew15 =

0.0590	0.0824	0.8586
0.0008	0.0580	0.9412
0.5860	0.4140	0.0000

Bnew15 =

0.1441	0.8559	0
0.0537	0.9463	0
1.0000	0.0000	0

Pinew16 =

0.0014	0.9986	0.0000

Anew16 =

0.2709	0.6928	0.0363
0.0013	0.2573	0.7414
0.5008	0.0962	0.4030

Bnew16 =

0.0374	0.9626	0
0.1156	0.8844	0
0.9638	0.0362	0

Pinew17 =

0.0000	1.0000	0.0000

Anew17 =

0.0884	0.0006	0.9110
0.0437	0.5336	0.4227
0.8639	0.1361	0.0000

Bnew17 =

0.0000	1.0000	0
0.0000	1.0000	0
1.0000	0.0000	0

Pinew18 =

0.0015	0.9985	0.0000

Anew18 =

0.0804	0.0969	0.8227
0.0236	0.5351	0.4413
0.6189	0.3811	0.0000

Bnew18 =

0.0000	1.0000	0
0.0000	1.0000	0
0.9520	0.0480	0

Pinew19 =

0.1129 0.4265 0.4606

Anew19 =

0.0990 0.1083 0.7927
0.0051 0.2954 0.6995
0.4448 0.4594 0.0958

Bnew19 =

1 0 0
1 0 0
1 0 0

Pinew20 =

0.0003 0.9997 0.0000

Anew20 =

0.0027 0.1782 0.8191
0.0008 0.1379 0.8613
0.5122 0.3342 0.1536

Bnew20 =

0.2104 0.7896 0
0.9300 0.0700 0
1.0000 0.0000 0

Pinew21 =

0.1129 0.4265 0.4606

Anew21 =

0.0990 0.1083 0.7927
0.0051 0.2954 0.6995
0.4448 0.4594 0.0958

Bnew21 =

1 0 0
1 0 0
1 0 0

Pinew22 =

0.1648 0.8352 0.0000

Anew22 =

0.0000 0.2190 0.7810
0.0000 0.5603 0.4397
0.6768 0.3232 0.0000

Bnew22 =

0.0000 0.8448 0.1552
0.0000 0.2753 0.7247
0.9787 0.0000 0.0213

Pinew23 =

0.1129 0.4265 0.4606

Anew23 =

0.0990 0.1083 0.7927
0.0051 0.2954 0.6995
0.4448 0.4594 0.0958

Bnew23 =

1 0 0
1 0 0
1 0 0

Pinew24 =

0.0021 0.9979 0.0000

Anew24 =

0.0015 0.0739 0.9246
0.0003 0.1196 0.8801
0.4720 0.4494 0.0786

Bnew24 =

0.3296 0 0.6704
0.9620 0 0.0380
1.0000 0 0.0000

Pinew25 =

0.0000 0.0000 1.0000

Anew25 =

0.0001 0.2709 0.7290
0.0000 0.4392 0.5608
0.6175 0.3825 0.0000

Bnew25 =

0.0000 0.1925 0.8075
0.0000 0.9983 0.0017
1.0000 0.0000 0.0000

Pinew26 =

0.2079 0.7921 0.0000

Anew26 =

0.0590 0.0824 0.8586
0.0008 0.0580 0.9412
0.5860 0.4140 0.0000

Bnew26 =

0.1441 0.8559 0
0.0537 0.9463 0
1.0000 0.0000 0

Pinew27 =

0.0014 0.9986 0.0000

Anew27 =

0.2709 0.6928 0.0363
0.0013 0.2573 0.7414
0.5008 0.0962 0.4030

Bnew27 =

0.0374 0.9626 0
0.1156 0.8844 0
0.9638 0.0362 0

Pinew28 =

0.0000 1.0000 0.0000

Anew28 =

0.0884 0.0006 0.9110
0.0437 0.5336 0.4227
0.8639 0.1361 0.0000

Bnew28 =

0.0000 1.0000 0
0.0000 1.0000 0
1.0000 0.0000 0

Pinew29 =

0.0015 0.9985 0.0000

Anew29 =

0.0804 0.0969 0.8227
0.0236 0.5351 0.4413
0.6189 0.3811 0.0000

Bnew29 =

0.0000 1.0000 0
0.0000 1.0000 0
0.9520 0.0480 0

Pinew30 =

0.1129 0.4265 0.4606

Anew30 =

0.0990 0.1083 0.7927
0.0051 0.2954 0.6995
0.4448 0.4594 0.0958

Bnew30 =

1 0 0
1 0 0
1 0 0

APPENDIX B: MatLab Program

```
function M=normalise(A)
% NORMALISE make the entries of an array sum to 1.
s=sum(A,1)
% sum(A,DIM) sums along the dimension DIM
M=A/s
```

```
function [T,Z] = mk_stochastic(T)
% MK_STOCHASTIC Ensure the argument is a stochastic matrix, i.e., the sum over
each row is 1.
Z = sum(T,2);
norm = repmat(Z, 1, size(T,2));
T = T ./ norm;
```

```
function S = sample_mc(prior, trans, len)
% SAMPLE_MC Generate random sequences from a Markov chain.
% STATE = SAMPLE_MC(PRIOR, TRANS, LEN) generates a sequence of length
LEN.
S = zeros(1,len);
S(1,1) = sample_discrete(prior);
  for t=2:len
    S(1,t) = sample_discrete(trans(S(1,t-1),:));
  end
```

```
function M = sample_discrete(prob,  c)

% Example: sample_discrete([0.8 0.2], 1, 10) generates a row vector of 10 random
integers from {1,2},
% where the prob. of be ing 1 is 0.8 and the prob of being 2 is 0.2.

n = length(prob);

if nargin == 1
  c = 1;

end

R = rand(1, c);
M = ones(1, c);
cumprob = cumsum(prob(:));

if n < c
  for i = 1:n-1
    M = M + (R > cumprob(i));
  end
else
  % loop over the smaller index - can be much faster if length(prob) >> r*c
  cumprob2 = cumprob(1:end-1);

    for j=1:c
      M(1,j) = sum(R(1,j) > cumprob2)+1;
    end

end
```

```
function Y = sample_cond_multinomial(X, M)


Y = zeros(size(X));
for i=min(X(:)):max(X(:))
  ndx = find(X==i);
  Y(ndx) = sample_discrete(M(i,:), 1);
end
```

```
function obs= dhmm_sample(initial_prob, transmat, obsmat, len)
% SAMPLE_DHMM Generate random sequences from a Hidden Markov Model with
discrete outputs.
%
% [obs, hidden] = sample_dhmm(initial_prob, transmat, obsmat, len)
% obs is an observation sequence of length len.

hidden = generateMC(initial_prob, transmat, len);
obs = fromXtoY(hidden, obsmat);
obs
```

```
function [a,P]=initialization(Pi,A,B,Y)
%Here, Pi is a row vector.
%initialization
 N=size(Pi,2)
 for i=1:N
     a(1,i)=Pi(i)*B(i,Y(1))
 end
 %induction
 time=size(Y,2)
 for i=2:time
     b=a*A
     for j=1:N
         a(i,j)=b(i-1,j)*B(j,Y(i))
     end
 end
 s=sum(a,2)
 %This is the probability that the observable squence Y heppens.
 probability=s(time)
 P=probability
```

```
function beta=findbeta(Pi,A,B,Y)
%Here, Pi is a row vector.
%initialization
 N=size(Pi,2)
 T=size(Y,2)
 for i=1:N
    beta(T,i)=1
 end

 for t=T-1:-1:1
    for i=1:N
       beta(t,i)=0
       for p=1:N
          s(p)=A(i,p)*B(p,Y(t+1))*beta(t+1,p)
       beta(t,i)=beta(t,i)+s(p)
       end
    end
 end
```

```
function xi=findxi(Pi,A,B,Y,alpha,beta,P)

N=size(Pi,2)
T=size(Y,2)
for t=1:T-1
    for i=1:N
        for j=1:N
            xi(i,j,t)=alpha(t,i)*A(i,j)*B(j,Y(t+1))*beta(t+1,j)
            xi(i,j,t)=xi(i,j,t)/P
        end
    end
end
```

```
function gamma=findgamma(Pi,Y,xi)
N=size(Pi,2)
T=size(Y,2)
for t=1:T-1
   for i=1:N
      gamma(t,i)=0
      for p=1:N
         gamma(t,i)=gamma(t,i)+xi(i,p,t)
      end
   end
end
```

```
function [Pinew,Anew,Bnew]=findlambda(Pi,B,Y,gamma,xi)
N=size(Pi,2)
M=size(B,2)
T=size(Y,2)
%Pinew
for i=1:N
   Pinew(i)=gamma(1,i)
end
%Anew
r=sum(gamma)
for i=1:N
   for j=1:N
      k(i,j)=0
      for t=1:T-1
         k(i,j)=k(i,j)+xi(i,j,t)
      end
   end
end
for i=1:N
   for j=1:N
      Anew(i,j)=k(i,j)/r(i)
   end
end
%Bnew
for j=1:N
   for k=1:M
      p(j,k)=0
      for t=1:T-1
         if Y(t)==k
            p(j,k)=p(j,k)+gamma(t,j)
         end
      end
   end
end

for j=1:N
   for k=1:M
     Bnew(j,k)=p(j,k)/r(j)
  end
end
```

```
function [Pinew, Anew, Bnew]=better(A,B,Pi,Y)
[Pinew, Anew, Bnew]=good(A,B,Pi,Y)
for i=1:10
   [Pinew, Anew, Bnew]=good(Anew,Bnew,Pinew,Y)
end
```

```
function norm=getnorm(A,B)
norm=0
N=size(A)
M=size(A,2)
for i=1:N
   for j=1:M
      norm=norm+(A(i,j)-B(i,j))^2
   end
end
```

VITA

Yang Liu was born in Shuangyashan, Heilongjiang Province, China on April 28, 1979. She grew up in Beijing, China, and graduated from Beijing Bayi Middle School in July 1997. In July 2002, she received a Bachelor of Science degree in Mathematics from the University of Science and Technology of China, in Hefei, Anhui Province, China. The following fall, she entered the University of Tennessee at Knoxville to begin work towards a Master of Science degree in Mathematics, which was awarded in August 2004.